THE ART OF COMPUTER PROGRAMMING

PRE-FASCICLE 3A

A DRAFT OF SECTION 7.2.1.3: GENERATING ALL COMBINATIONS

DONALD E. KNUTH *Stanford University*

ADDISON–WESLEY

Internet page `http://www-cs-faculty.stanford.edu/~knuth/taocp.html` contains current information about this book and related books.

See also `http://www-cs-faculty.stanford.edu/~knuth/sgb.html` for information about *The Stanford GraphBase*, including downloadable software for dealing with the graphs used in many of the examples in Chapter 7.

See also `http://www-cs-faculty.stanford.edu/~knuth/mmixware.html` for downloadable software to simulate the MMIX computer.

Zeroth printing (revision 11), 31 March 2005

PREFACE

[The Art of Combinations] has a relation
to almost every species of useful knowledge
that the mind of man can be employed upon.
— JAMES BERNOULLI, *Ars Conjectandi* (1713)

THIS BOOKLET contains draft material that I'm circulating to experts in the field, in hopes that they can help remove its most egregious errors before too many other people see it. I am also, however, posting it on the Internet for courageous and/or random readers who don't mind the risk of reading a few pages that have not yet reached a very mature state. *Beware:* This material has not yet been proofread as thoroughly as the manuscripts of Volumes 1, 2, and 3 were at the time of their first printings. And those carefully-checked volumes, alas, were subsequently found to contain thousands of mistakes.

Given this caveat, I hope that my errors this time will not be so numerous and/or obtrusive that you will be discouraged from reading the material carefully. I did try to make it both interesting and authoritative, as far as it goes. But the field is so vast, I cannot hope to have surrounded it enough to corral it completely. Therefore I beg you to let me know about any deficiencies you discover.

To put the material in context, this is Section 7.2.1.3 of a long, long chapter on combinatorial algorithms. Chapter 7 will eventually fill three volumes (namely Volumes 4A, 4B, and 4C), assuming that I'm able to remain healthy. It will begin with a short review of graph theory, with emphasis on some highlights of significant graphs in The Stanford GraphBase, from which I will be drawing many examples. Then comes Section 7.1, which deals with the topic of bitwise manipulations. (I drafted about 60 pages about that subject in 1977, but those pages need extensive revision; meanwhile I've decided to work for awhile on the material that follows it, so that I can get a better feel for how much to cut.) Section 7.2 is about generating all possibilities, and it begins with Section 7.2.1: Generating Basic Combinatorial Patterns — which, in turn, begins with Section 7.2.1.1, "Generating all n-tuples," and Section 7.2.1.2, "Generating all permutations." (Readers of the present booklet should have already looked at those sections, drafts of which are available as Pre-Fascicles 2A and 2B.) The stage is now set for the main contents of this booklet, Section 7.2.1.3: "Generating all combinations." Then will come Section 7.2.1.4 (about partitions), etc. Section 7.2.2 will deal with backtracking in general. And so it will go on, if all goes well; an outline of the entire Chapter 7 as currently envisaged appears on the `taocp` webpage that is cited on page ii.

Even the apparently lowly topic of combination generation turns out to be surprisingly rich, with ties to Sections 1.2.1, 1.2.4, 1.2.6, 2.3.2, 2.3.4.2, 3.4.2, 4.3.2, 4.6.1, 4.6.2, 5.1.2, 5.4.1, 5.4.2, 6.1, and 6.3 of the first three volumes. I strongly believe in building up a firm foundation, so I have discussed this topic much more thoroughly than I will be able to do with material that is newer or less basic. To my surprise, I came up with 110 exercises, even though — believe it or not — I had to eliminate quite a bit of the interesting material that appears in my files.

Some of the things presented are new, to the best of my knowledge, although I will not be at all surprised to learn that my own little "discoveries" have been discovered before. Please look, for example, at the exercises that I've classed as research problems (rated with difficulty level 46 or higher), namely exercises 53, 56, 67, and 83; I've also implicitly posed additional unsolved questions in the answers to exercises 59, 63, 101, 105, and 109. Are those problems still open? Please let me know if you know of a solution to any of these intriguing questions. And of course if no solution is known today but you do make progress on any of them in the future, I hope you'll let me know.

I urgently need your help also with respect to some exercises that I made up as I was preparing this material. I certainly don't like to get credit for things that have already been published by others, and most of these results are quite natural "fruits" that were just waiting to be "plucked." Therefore please tell me if you know who I should have credited, with respect to the ideas found in exercises 9, 18, 19, 20, 26, 27, 28, 30, 31, 32, 33, 34, 35, 36, 37, 41, 42, 43, 44, 45, 48, 51, 59, 62, 63, 64, 65, 66, 69, 79, 82(b–f), 85, 86, 87, 93, and/or 110.

I shall happily pay a finder's fee of $2.56 for each error in this draft when it is first reported to me, whether that error be typographical, technical, or historical. The same reward holds for items that I forgot to put in the index. And valuable suggestions for improvements to the text are worth 32¢ each. (Furthermore, if you find a better solution to an exercise, I'll actually reward you with immortal glory instead of mere money, by publishing your name in the eventual book:–)

Cross references to yet-unwritten material sometimes appear as '00'; this impossible value is a placeholder for the actual numbers to be supplied later.

Happy reading!

Stanford, California D. E. K.
13 June 2002

7.2.1.3. Generating all combinations. Combinatorial mathematics is often described as "the study of permutations, combinations, etc.," so we turn our attention now to combinations. A *combination of n things, taken t at a time*, often called simply a t-combination of n things, is a way to select a subset of size t from a given set of size n. We know from Eq. 1.2.6–(2) that there are exactly $\binom{n}{t}$ ways to do this; and we learned in Section 3.4.2 how to choose t-combinations at random.

Selecting t of n objects is equivalent to choosing the $n-t$ elements not selected. We will emphasize this symmetry by letting

$$n \;=\; s + t \tag{1}$$

throughout our discussion, and we will often refer to a t-combination of n things as an "(s,t)-combination." Thus, an (s,t)-combination is a way to subdivide $s+t$ objects into two collections of sizes s and t.

If I ask how many combinations of 21 can be taken out of 25,
I do in effect ask how many combinations of 4 may be taken.
For there are just as many ways of taking 21 as there are of leaving 4.
— AUGUSTUS DE MORGAN, *An Essay on Probabilities* (1838)

There are two main ways to represent (s,t)-combinations: We can list the elements $c_t \ldots c_2 c_1$ that have been selected, or we can work with binary strings $a_{n-1} \ldots a_1 a_0$ for which

$$a_{n-1} + \cdots + a_1 + a_0 \;=\; t. \tag{2}$$

The latter representation has s 0s and t 1s, corresponding to elements that are unselected or selected. The list representation $c_t \ldots c_2 c_1$ tends to work out best if we let the elements be members of the set $\{0, 1, \ldots, n-1\}$ and if we list them in *decreasing* order:

$$n > c_t > \cdots > c_2 > c_1 \ge 0. \tag{3}$$

Binary notation connects these two representations nicely, because the item list $c_t \ldots c_2 c_1$ corresponds to the sum

$$2^{c_t} + \cdots + 2^{c_2} + 2^{c_1} \;=\; \sum_{k=0}^{n-1} a_k 2^k \;=\; (a_{n-1} \ldots a_1 a_0)_2. \tag{4}$$

Of course we could also list the positions $b_s \ldots b_2 b_1$ of the 0s in $a_{n-1} \ldots a_1 a_0$, where

$$n > b_s > \cdots > b_2 > b_1 \geq 0. \tag{5}$$

Combinations are important not only because subsets are omnipresent in mathematics but also because they are equivalent to many other configurations. For example, every (s,t)-combination corresponds to a combination of $s+1$ things taken t at a time *with repetitions permitted*, also called a *multicombination*, namely a sequence of integers $d_t \ldots d_2 d_1$ with

$$s \geq d_t \geq \cdots \geq d_2 \geq d_1 \geq 0. \tag{6}$$

One reason is that $d_t \ldots d_2 d_1$ solves (6) if and only if $c_t \ldots c_2 c_1$ solves (3), where

$$c_t = d_t + t - 1, \quad \ldots, \quad c_2 = d_2 + 1, \quad c_1 = d_1 \tag{7}$$

(see exercise 1.2.6–60). And there is another useful way to relate combinations with repetition to ordinary combinations, suggested by Solomon Golomb [*AMM* **75** (1968), 530–531], namely to define

$$e_j \;=\; \begin{cases} c_j, & \text{if } c_j \leq s; \\ e_{c_j - s}, & \text{if } c_j > s. \end{cases} \tag{8}$$

In this form the numbers $e_t \ldots e_1$ don't necessarily appear in descending order, but the multiset $\{e_1, e_2, \ldots, e_t\}$ is equal to $\{c_1, c_2, \ldots, c_t\}$ if and only if $\{e_1, e_2, \ldots, e_t\}$ is a set. (See Table 1 and exercise 1.)

An (s,t)-combination is also equivalent to a *composition* of $n+1$ into $t+1$ parts, namely an ordered sum

$$n + 1 \;=\; p_t + \cdots + p_1 + p_0, \qquad \text{where } p_t, \ldots, p_1, p_0 \geq 1. \tag{9}$$

The connection with (3) is now

$$p_t = n - c_t, \quad p_{t-1} = c_t - c_{t-1}, \quad \ldots, \quad p_1 = c_2 - c_1, \quad p_0 = c_1 + 1. \tag{10}$$

Equivalently, if $q_j = p_j - 1$, we have

$$s \;=\; q_t + \cdots + q_1 + q_0, \qquad \text{where } q_t, \ldots, q_1, q_0 \geq 0, \tag{11}$$

a composition of s into $t+1$ *nonnegative* parts, related to (6) by setting

$$q_t = s - d_t, \quad q_{t-1} = d_t - d_{t-1}, \quad \ldots, \quad q_1 = d_2 - d_1, \quad q_0 = d_1. \tag{12}$$

Furthermore it is easy to see that an (s,t)-combination is equivalent to a path of length $s+t$ from corner to corner of an $s \times t$ grid, because such a path contains s vertical steps and t horizontal steps. Thus, combinations can be studied in at least eight different guises. Table 1 illustrates all $\binom{6}{3} = 20$ possibilities in the case $s = t = 3$.

These cousins of combinations might seem rather bewildering at first glance, but most of them can be understood directly from the binary representation $a_{n-1} \ldots a_1 a_0$. Consider, for example, the "random" bit string

$$a_{23} \ldots a_1 a_0 \;=\; 011001001000011111101101, \tag{13}$$

Table 1

THE $(3,3)$-COMBINATIONS AND THEIR EQUIVALENTS

$a_5a_4a_3a_2a_1a_0$	$b_3b_2b_1$	$c_3c_2c_1$	$d_3d_2d_1$	$e_3e_2e_1$	$p_3p_2p_1p_0$	$q_3q_2q_1q_0$	path
000111	543	210	000	210	4111	3000	
001011	542	310	100	310	3211	2100	
001101	541	320	110	320	3121	2010	
001110	540	321	111	321	3112	2001	
010011	532	410	200	010	2311	1200	
010101	531	420	210	020	2221	1110	
010110	530	421	211	121	2212	1101	
011001	521	430	220	030	2131	1020	
011010	520	431	221	131	2122	1011	
011100	510	432	222	232	2113	1002	
100011	432	510	300	110	1411	0300	
100101	431	520	310	220	1321	0210	
100110	430	521	311	221	1312	0201	
101001	421	530	320	330	1231	0120	
101010	420	531	321	331	1222	0111	
101100	410	532	322	332	1213	0102	
110001	321	540	330	000	1141	0030	
110010	320	541	331	111	1132	0021	
110100	310	542	332	222	1123	0012	
111000	210	543	333	333	1114	0003	

which has $s = 11$ zeros and $t = 13$ ones, hence $n = 24$. The dual combination $b_s \ldots b_1$ lists the positions of the zeros, namely

$$23\ 20\ 19\ 17\ 16\ 14\ 13\ 12\ 11\ 4\ 1,$$

because the leftmost position is $n - 1$ and the rightmost is 0. The primal combination $c_t \ldots c_1$ lists the positions of the ones, namely

$$22\ 21\ 18\ 15\ 10\ 9\ 8\ 7\ 6\ 5\ 3\ 2\ 0.$$

The corresponding multicombination $d_t \ldots d_1$ lists the number of 0s to the right of each 1:

$$10\ 10\ 8\ 6\ 2\ 2\ 2\ 2\ 2\ 2\ 1\ 1\ 0.$$

The composition $p_t \ldots p_0$ lists the distances between consecutive 1s, if we imagine additional 1s at the left and the right:

$$2\ 1\ 3\ 3\ 5\ 1\ 1\ 1\ 1\ 1\ 2\ 1\ 2\ 1.$$

And the nonnegative composition $q_t \ldots q_0$ counts how many 0s appear between "fenceposts" represented by 1s:

$$1\ 0\ 2\ 2\ 4\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0;$$

thus we have

$$a_{n-1} \ldots a_1 a_0 = 0^{q_t} 1 0^{q_{t-1}} 1 \ldots 1 0^{q_1} 1 0^{q_0}. \qquad (14)$$

The paths in Table 1 also have a simple interpretation (see exercise 2).

Lexicographic generation. Table 1 shows combinations $a_{n-1}\ldots a_1a_0$ and $c_t\ldots c_1$ in lexicographic order, which is also the lexicographic order of $d_t\ldots d_1$. Notice that the dual combinations $b_s\ldots b_1$ and the corresponding compositions $p_t\ldots p_0$, $q_t\ldots q_0$ then appear in *reverse* lexicographic order.

Lexicographic order usually suggests the most convenient way to generate combinatorial configurations. Indeed, Algorithm 7.2.1.2L already solves the problem for combinations in the form $a_{n-1}\ldots a_1a_0$, since (s,t)-combinations in bitstring form are the same as permutations of the multiset $\{s\cdot 0, t\cdot 1\}$. That general-purpose algorithm can be streamlined in obvious ways when it is applied to this special case. (See also exercise 7.1–00, which presents a remarkable sequence of seven bitwise operations that will convert any given binary number $(a_{n-1}\ldots a_1a_0)_2$ to the lexicographically next t-combination, assuming that n does not exceed the computer's word length.)

Let's focus, however, on generating combinations in the other principal form $c_t\ldots c_2c_1$, which is more directly relevant to the ways in which combinations are often needed, and which is more compact than the bit strings when t is small compared to n. In the first place we should keep in mind that a simple sequence of nested loops will do the job nicely when t is very small. For example, when $t=3$ the following instructions suffice:

For $c_3 = 2, 3, \ldots, n-1$ (in this order) do the following:
 For $c_2 = 1, 2, \ldots, c_3-1$ (in this order) do the following:
  For $c_1 = 0, 1, \ldots, c_2-1$ (in this order) do the following:
   Visit the combination $c_3c_2c_1$. (15)

(See the analogous situation in 7.2.1.1–(3).)

On the other hand when t is variable or not so small, we can generate combinations lexicographically by following the general recipe discussed after Algorithm 7.2.1.2L, namely to find the rightmost element c_j that can be increased and then to set the subsequent elements $c_{j-1}\ldots c_1$ to their smallest possible values:

Algorithm L (*Lexicographic combinations*). This algorithm generates all t-combinations $c_t\ldots c_2c_1$ of the n numbers $\{0,1,\ldots,n-1\}$, given $n\ge t\ge 0$. Additional variables c_{t+1} and c_{t+2} are used as sentinels.

L1. [Initialize.] Set $c_j \leftarrow j-1$ for $1\le j\le t$; also set $c_{t+1}\leftarrow n$ and $c_{t+2}\leftarrow 0$.

L2. [Visit.] Visit the combination $c_t\ldots c_2c_1$.

L3. [Find j.] Set $j\leftarrow 1$. Then, while $c_j+1=c_{j+1}$, set $c_j\leftarrow j-1$ and $j\leftarrow j+1$; repeat until $c_j+1\ne c_{j+1}$.

L4. [Done?] Terminate the algorithm if $j>t$.

L5. [Increase c_j.] Set $c_j\leftarrow c_j+1$ and return to L2. ▮

The running time of this algorithm is not difficult to analyze. Step L3 sets $c_j\leftarrow j-1$ just after visiting a combination for which $c_{j+1}=c_1+j$, and the number of such combinations is the number of solutions to the inequalities

$$n > c_t > \cdots > c_{j+1} \ge j; \tag{16}$$

but this formula is equivalent to a $(t-j)$-combination of the $n-j$ objects $\{n-1,\ldots,j\}$, so the assignment $c_j \leftarrow j-1$ occurs exactly $\binom{n-j}{t-j}$ times. Summing for $1 \le j \le t$ tells us that the loop in step L3 is performed

$$\binom{n-1}{t-1}+\binom{n-2}{t-2}+\cdots+\binom{n-t}{0} = \binom{n-1}{s}+\binom{n-2}{s}+\cdots+\binom{s}{s} = \binom{n}{s+1} \qquad (17)$$

times altogether, or an average of

$$\binom{n}{s+1}\Big/\binom{n}{t} = \frac{n!}{(s+1)!\,(t-1)!}\Big/\frac{n!}{s!\,t!} = \frac{t}{s+1} \qquad (18)$$

times per visit. This ratio is less than 1 when $t \le s$, so Algorithm L is quite efficient in such cases.

But the quantity $t/(s+1)$ can be embarrassingly large when t is near n and s is small. Indeed, Algorithm L occasionally sets $c_j \leftarrow j-1$ needlessly, at times when c_j already equals $j-1$. Further scrutiny reveals that we need not always search for the index j that is needed in steps L4 and L5, since the correct value of j can often be predicted from the actions just taken. For example, after we have increased c_4 and reset $c_3c_2c_1$ to their starting values 210, the next combination will inevitably increase c_3. These observations lead to a tuned-up version of the algorithm:

Algorithm T (*Lexicographic combinations*). This algorithm is like Algorithm L, but faster. It also assumes, for convenience, that $t < n$.

T1. [Initialize.] Set $c_j \leftarrow j-1$ for $1 \le j \le t$; then set $c_{t+1} \leftarrow n$, $c_{t+2} \leftarrow 0$, and $j \leftarrow t$.

T2. [Visit.] (At this point j is the smallest index such that $c_{j+1} > j$.) Visit the combination $c_t \ldots c_2c_1$. Then, if $j > 0$, set $x \leftarrow j$ and go to step T6.

T3. [Easy case?] If $c_1 + 1 < c_2$, set $c_1 \leftarrow c_1 + 1$ and return to T2. Otherwise set $j \leftarrow 2$.

T4. [Find j.] Set $c_{j-1} \leftarrow j-2$ and $x \leftarrow c_j + 1$. If $x = c_{j+1}$, set $j \leftarrow j+1$ and repeat this step until $x \ne c_{j+1}$.

T5. [Done?] Terminate the algorithm if $j > t$.

T6. [Increase c_j.] Set $c_j \leftarrow x$, $j \leftarrow j-1$, and return to T2. ▮

Now $j = 0$ in step T2 if and only if $c_1 > 0$, so the assignments in step T4 are never redundant. Exercise 6 carries out a complete analysis of Algorithm T.

Notice that the parameter n appears only in the initialization steps L1 and T1, not in the principal parts of Algorithms L and T. Thus we can think of the process as generating the first $\binom{n}{t}$ combinations of an *infinite* list, which depends only on t. This simplification arises because the list of t-combinations for $n+1$ things begins with the list for n things, under our conventions; we have been using lexicographic order on the decreasing sequences $c_t \ldots c_1$ for this very reason, instead of working with the increasing sequences $c_1 \ldots c_t$.

Derrick Lehmer noticed another pleasant property of Algorithms L and T [*Applied Combinatorial Mathematics*, edited by E. F. Beckenbach (1964), 27–30]:

Theorem L. *The combination $c_t \ldots c_2 c_1$ is visited after exactly*

$$\binom{c_t}{t} + \cdots + \binom{c_2}{2} + \binom{c_1}{1} \tag{19}$$

other combinations have been visited.

Proof. There are $\binom{c_k}{k}$ combinations $c'_t \ldots c'_2 c'_1$ with $c'_j = c_j$ for $t \geq j > k$ and $c'_k < c_k$, namely $c_t \ldots c_{k+1}$ followed by the k-combinations of $\{0, \ldots, c_k - 1\}$. ▌

When $t = 3$, for example, the numbers

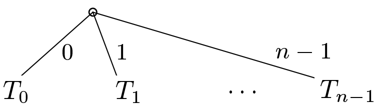
$$\tbinom{2}{3}+\tbinom{1}{2}+\tbinom{0}{1},\ \tbinom{3}{3}+\tbinom{1}{2}+\tbinom{0}{1},\ \tbinom{3}{3}+\tbinom{2}{2}+\tbinom{0}{1},\ \ldots,\ \tbinom{5}{3}+\tbinom{4}{2}+\tbinom{3}{1}$$

that correspond to the combinations $c_3 c_2 c_1$ in Table 1 simply run through the sequence 0, 1, 2, ..., 19. Theorem L gives us a nice way to understand the *combinatorial number system* of degree t, which represents every nonnegative integer N uniquely in the form

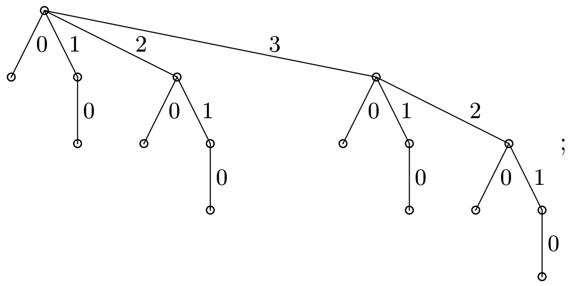
$$N = \binom{n_t}{t} + \cdots + \binom{n_2}{2} + \binom{n_1}{1}, \qquad n_t > \cdots > n_2 > n_1 \geq 0. \tag{20}$$

[See Ernesto Pascal, *Giornale di Matematiche* **25** (1887), 45–49.]

Binomial trees. The family of trees T_n defined by

$T_0 = \circ$, $\qquad T_n =$  for $n > 0$, (21)

arises in several important contexts and sheds further light on combination generation. For example, T_4 is

 ; (22)

and T_5, rendered more artistically, appears as the frontispiece to Volume 1 of this series of books.

Notice that T_n is like T_{n-1}, except for an additional copy of T_{n-1}; therefore T_n has 2^n nodes altogether. Furthermore, the number of nodes on level t is the binomial coefficient $\binom{n}{t}$; this fact accounts for the name "binomial tree." Indeed, the sequence of labels encountered on the path from the root to each node on level t defines a combination $c_t \ldots c_1$, and all combinations occur in lexicographic order from left to right. Thus, Algorithms L and T can be regarded as procedures to traverse the nodes on level t of the binomial tree T_n.

The infinite binomial tree T_∞ is obtained by letting $n \to \infty$ in (21). The root of this tree has infinitely many branches, but every node except for the overall root at level 0 is the root of a finite binomial subtree. All possible t-combinations appear in lexicographic order on level t of T_∞.

Let's get more familiar with binomial trees by considering all possible ways to pack a rucksack. More precisely, suppose we have n items that take up respectively $w_{n-1}, \ldots, w_1, w_0$ units of capacity, where

$$w_{n-1} \geq \cdots \geq w_1 \geq w_0; \tag{23}$$

we want to generate all binary vectors $a_{n-1} \ldots a_1 a_0$ such that

$$a \cdot w = a_{n-1}w_{n-1} + \cdots + a_1 w_1 + a_0 w_0 \leq N, \tag{24}$$

where N is the total capacity of a rucksack. Equivalently, we want to find all subsets C of $\{0, 1, \ldots, n-1\}$ such that $w(C) = \sum_{c \in C} w_c \leq N$; such subsets will be called *feasible*. We will write a feasible subset as $c_1 \ldots c_t$, where $c_1 > \cdots > c_t \geq 0$, numbering the subscripts differently from the convention of (3) above because t is variable in this problem.

Every feasible subset corresponds to a node of T_n, and our goal is to visit each feasible node. Clearly the parent of every feasible node is feasible, and so is the left sibling, if any; therefore a simple tree exploration procedure works well:

Algorithm F (*Filling a rucksack*). This algorithm generates all feasible ways $c_1 \ldots c_t$ to fill a rucksack, given $w_{n-1}, \ldots, w_1, w_0$, and N. We let $\delta_j = w_j - w_{j-1}$ for $1 \leq j < n$.

F1. [Initialize.] Set $t \leftarrow 0$, $c_0 \leftarrow n$, and $r \leftarrow N$.

F2. [Visit.] Visit the combination $c_1 \ldots c_t$, which uses $N - r$ units of capacity.

F3. [Try to add w_0.] If $c_t > 0$ and $r \geq w_0$, set $t \leftarrow t + 1$, $c_t \leftarrow 0$, $r \leftarrow r - w_0$, and return to F2.

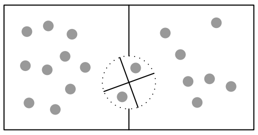
F4. [Try to increase c_t.] Terminate if $t = 0$. Otherwise, if $c_{t-1} > c_t + 1$ and $r \geq \delta_{c_t+1}$, set $c_t \leftarrow c_t + 1$, $r \leftarrow r - \delta_{c_t}$, and return to F2.

F5. [Remove c_t.] Set $r \leftarrow r + w_{c_t}$, $t \leftarrow t - 1$, and return to F4. ▮

Notice that the algorithm implicitly visits nodes of T_n in preorder, skipping over unfeasible subtrees. An element $c > 0$ is placed in the rucksack, if it fits, just after the procedure has explored all possibilities using element $c - 1$ in its place. The running time is proportional to the number of feasible combinations visited (see exercise 20).

Incidentally, the classical "knapsack problem" of operations research is different: It asks for a feasible subset C such that $v(C) = \sum_{c \in C} v(c)$ is maximum, where each item c has been assigned a value $v(c)$. Algorithm F is not a particularly good way to solve that problem, because it often considers cases that could be ruled out. For example, if C and C' are subsets of $\{1, \ldots, n-1\}$ with $w(C) \leq w(C') \leq N - w_0$ and $v(C) \geq v(C')$, Algorithm F will examine both $C \cup \{0\}$ and $C' \cup \{0\}$, but the latter subset will never improve the maximum. We will consider methods for the classical knapsack problem later; Algorithm F is intended only for situations when *all* of the feasible possibilities are potentially relevant.

Gray codes for combinations. Instead of merely generating all combinations, we often prefer to visit them in such a way that each one is obtained by making only a small change to its predecessor.



For example, we can ask for what Nijenhuis and Wilf have called a "revolving door algorithm": Imagine two rooms that contain respectively s and t people, with a revolving door between them. Whenever a person goes into the opposite room, somebody else comes out. Can we devise a sequence of moves so that each (s,t)-combination occurs exactly once?

The answer is yes, and in fact a huge number of such patterns exist. For example, it turns out that if we examine all n-bit strings $a_{n-1}\ldots a_1a_0$ in the well-known order of Gray binary code (Section 7.2.1.1), but select only those that have exactly s 0s and t 1s, the resulting strings form a revolving-door code.

Here's the proof: Gray binary code is defined by the recurrence $\Gamma_n = 0\Gamma_{n-1}$, $1\Gamma^R_{n-1}$ of 7.2.1.1–(5), so its (s,t) subsequence satisfies the recurrence

$$\Gamma_{st} \;=\; 0\Gamma_{(s-1)t},\; 1\Gamma^R_{s(t-1)} \tag{25}$$

when $st > 0$. We also have $\Gamma_{s0} = 0^s$ and $\Gamma_{0t} = 1^t$. Therefore it is clear by induction that Γ_{st} begins with 0^s1^t and ends with 10^s1^{t-1} when $st > 0$. The transition at the comma in (25) is from the last element of $0\Gamma_{(s-1)t}$ to the last element of $1\Gamma_{s(t-1)}$, namely from $010^{s-1}1^{t-1} = 010^{s-1}11^{t-2}$ to $110^s1^{t-2} = 110^{s-1}01^{t-2}$ when $t \ge 2$, and this satisfies the revolving-door constraint. The case $t = 1$ also checks out. For example, Γ_{33} is given by the columns of

000111	011010	110001	101010	
001101	011100	110010	101100	
001110	010101	110100	100101	(26)
001011	010110	111000	100110	
011001	010011	101001	100011	

and Γ_{23} can be found in the first two columns of this array. One more turn of the door takes the last element into the first. [These properties of Γ_{st} were discovered by D. T. Tang and C. N. Liu, *IEEE Trans.* **C-22** (1973), 176–180; a loopless implementation was presented by J. R. Bitner, G. Ehrlich, and E. M. Reingold, *CACM* **19** (1976), 517–521.]

When we convert the bit strings $a_5a_4a_3a_2a_1a_0$ in (26) to the corresponding index-list forms $c_3c_2c_1$, a striking pattern becomes evident:

210	431	540	531	
320	432	541	532	
321	420	542	520	(27)
310	421	543	521	
430	410	530	510	

The first components c_3 occur in increasing order; but for each fixed value of c_3, the values of c_2 occur in *decreasing* order. And for fixed c_3c_2, the values of c_1 are again increasing. The same is true in general: *All combinations* $c_t\ldots c_2c_1$

appear in lexicographic order of

$$(c_t,\ -c_{t-1},\ c_{t-2},\ \ldots,\ (-1)^{t-1}c_1) \tag{28}$$

in the revolving-door Gray code Γ_{st}. This property follows by induction, because (25) becomes

$$\Gamma_{st} \;=\; \Gamma_{(s-1)t},\ (s{+}t{-}1)\Gamma^R_{s(t-1)} \tag{29}$$

for $st > 0$ when we use index-list notation instead of bitstring notation. Consequently the sequence can be generated efficiently by the following algorithm due to W. H. Payne [see *ACM Trans. Math. Software* **5** (1979), 163–172]:

Algorithm R (*Revolving-door combinations*). This algorithm generates all t-combinations $c_t \ldots c_2c_1$ of $\{0, 1, \ldots, n-1\}$ in lexicographic order of the alternating sequence (28), assuming that $n > t > 1$. Step R3 has two variants, depending on whether t is even or odd.

R1. [Initialize.] Set $c_j \leftarrow j-1$ for $t \ge j \ge 1$, and $c_{t+1} \leftarrow n$.

R2. [Visit.] Visit the combination $c_t \ldots c_2c_1$.

R3. [Easy case?] If t is odd: If $c_1 + 1 < c_2$, increase c_1 by 1 and return to R2, otherwise set $j \leftarrow 2$ and go to R4. If t is even: If $c_1 > 0$, decrease c_1 by 1 and return to R2, otherwise set $j \leftarrow 2$ and go to R5.

R4. [Try to decrease c_j.] (At this point $c_j = c_{j-1} + 1$.) If $c_j \ge j$, set $c_j \leftarrow c_{j-1}$, $c_{j-1} \leftarrow j-2$, and return to R2. Otherwise increase j by 1.

R5. [Try to increase c_j.] (At this point $c_{j-1} = j-2$.) If $c_j + 1 < c_{j+1}$, set $c_{j-1} \leftarrow c_j$, $c_j \leftarrow c_j + 1$, and return to R2. Otherwise increase j by 1, and go to R4 if $j \le t$. ▮

Exercises 21–25 explore further properties of this interesting sequence. One of them is a nice companion to Theorem L: *The combination $c_tc_{t-1}\ldots c_2c_1$ is visited by Algorithm R after exactly*

$$N = \binom{c_t{+}1}{t} - \binom{c_{t-1}{+}1}{t{-}1} + \cdots + (-1)^t\binom{c_2{+}1}{2} - (-1)^t\binom{c_1{+}1}{1} - [t\ \text{odd}] \tag{30}$$

other combinations have been visited. We may call this the representation of N in the "alternating combinatorial number system" of degree t; one consequence, for example, is that every positive integer has a unique representation of the form $N = \binom{a}{3} - \binom{b}{2} + \binom{c}{1}$ with $a > b > c > 0$. Algorithm R tells us how to add 1 to N in this system.

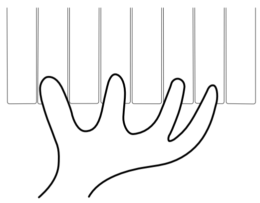
Although the strings of (26) and (27) are not in lexicographic order, they are examples of a more general concept called *genlex order*, a name coined by Timothy Walsh. A sequence of strings $\alpha_1, \ldots, \alpha_N$ is said to be in genlex order when all strings with a common prefix occur consecutively. For example, all 3-combinations that begin with 53 appear together in (27).

Genlex order means that the strings can be arranged in a trie structure, as in Fig. 31 of Section 6.3, but with the children of each node ordered arbitrarily. When a trie is traversed in any order such that each node is visited just before or just after its descendants, all nodes with a common prefix — that is, all nodes of a subtrie — appear consecutively. This principle makes genlex order convenient, because it corresponds to recursive generation schemes. Many of the algorithms we have seen for generating n-tuples have therefore produced their results in some version of genlex order; similarly, the method of "plain changes" (Algorithm 7.2.1.2P) visits permutations in a genlex order of the corresponding inversion tables.

The revolving-door method of Algorithm R is a genlex routine that changes only one element of the combination at each step. But it isn't totally satisfactory, because it frequently must change two of the indices c_j simultaneously, in order to preserve the condition $c_t > \cdots > c_2 > c_1$. For example, Algorithm R changes 210 into 320, and (27) includes nine such "crossing" moves.

The source of this defect can be traced to our proof that (25) satisfies the revolving-door property: We observed that the string $010^{s-1}11^{t-2}$ is followed by $110^{s-1}01^{t-2}$ when $t \ge 2$. Hence the recursive construction Γ_{st} involves transitions of the form $110^a0 \leftrightarrow 010^a1$, when a substring like 11000 is changed to 01001 or vice versa; the two 1s cross each other.

A Gray path for combinations is said to be *homogeneous* if it changes only one of the indices c_j at each step. A homogeneous scheme is characterized in bitstring form by having only transitions of the forms $10^a \leftrightarrow 0^a1$ within strings, for $a \ge 1$, when we pass from one string to the next. With a homogeneous scheme we can, for example, play all t-note chords on an n-note keyboard by moving only one finger at a time.



A slight modification of (25) yields a genlex scheme for (s,t)-combinations that is pleasantly homogeneous. The basic idea is to construct a sequence that begins with 0^s1^t and ends with 1^t0^s, and the following recursion suggests itself almost immediately: Let $K_{s0} = 0^s$, $K_{0t} = 1^t$, $K_{s(-1)} = \emptyset$, and

$$K_{st} \;=\; 0K_{(s-1)t},\; 10K^R_{(s-1)(t-1)},\; 11K_{s(t-2)} \quad \text{for } st > 0. \tag{31}$$

At the commas of this sequence we have 01^t0^{s-1} followed by $101^{t-1}0^{s-1}$, and 10^s1^{t-1} followed by 110^s1^{t-2}; both of these transitions are homogeneous, although the second one requires the 1 to jump across s 0s. The combinations K_{33} for $s = t = 3$ are

000111	010101	101100	100011	
001011	010011	101001	110001	
001101	011001	101010	110010	(32)
001110	011010	100110	110100	
010110	011100	100101	111000	

in bitstring form, and the corresponding "finger patterns" are

210	420	532	510	
310	410	530	540	
320	430	531	541	(33)
321	431	521	542	
421	432	520	543.	

When a homogeneous scheme for ordinary combinations $c_t \ldots c_1$ is converted to the corresponding scheme (6) for combinations with repetitions $d_t \ldots d_1$, it retains the property that only one of the indices d_j changes at each step. And when it is converted to the corresponding schemes (9) or (11) for compositions $p_t \ldots p_0$ or $q_t \ldots q_0$, only two (adjacent) parts change when c_j changes.

Near-perfect schemes. But we can do even better! All (s,t)-combinations can be generated by a sequence of strongly homogeneous transitions that are either $01 \leftrightarrow 10$ or $001 \leftrightarrow 100$. In other words, we can insist that each step causes a single index c_j to change by at most 2. Let's call such generation schemes *near-perfect.*

Imposing such strong conditions actually makes it fairly easy to discover near-perfect schemes, because comparatively few choices are available. Indeed, if we restrict ourselves to genlex methods that are near-perfect on n-bit strings, T. A. Jenkyns and D. McCarthy observed that all such methods can be easily characterized [*Ars Combinatoria* **40** (1995), 153–159]:

Theorem N. *If $st > 0$, there are exactly $2s$ near-perfect ways to list all (s,t)-combinations in a genlex order. In fact, when $1 \le a \le s$, there is exactly one such listing, N_{sta}, that begins with 1^t0^s and ends with $0^a1^t0^{s-a}$; the other s possibilities are the reverse lists, N^R_{sta}.*

Proof. The result certainly holds when $s = t = 1$; otherwise we use induction on $s+t$. The listing N_{sta}, if it exists, must have the form $1X_{s(t-1)}, 0Y_{(s-1)t}$ for some near-perfect genlex listings $X_{s(t-1)}$ and $Y_{(s-1)t}$. If $t = 1$, $X_{s(t-1)}$ is the single string 0^s; hence $Y_{(s-1)t}$ must be $N_{(s-1)1(a-1)}$ if $a > 1$, and it must be $N^R_{(s-1)11}$ if $a = 1$. On the other hand if $t > 1$, the near-perfect condition implies that the last string of $X_{s(t-1)}$ cannot begin with 1; hence $X_{s(t-1)} = N_{s(t-1)b}$ for some b. If $a > 1$, $Y_{(s-1)t}$ must be $N_{(s-1)t(a-1)}$, hence b must be 1; similarly, b must be 1 if $s = 1$. Otherwise we have $a = 1 < s$, and this forces $Y_{(s-1)t} = N^R_{(s-1)tc}$ for some c. The transition from $10^b1^{t-1}0^{s-b}$ to $0^{c+1}1^t0^{s-1-c}$ is near-perfect only if $c = 1$ and $b = 2$. ▮

The proof of Theorem N yields the following recursive formulas when $st > 0$:

$$N_{sta} = \begin{cases} 1N_{s(t-1)1},\ 0N_{(s-1)t(a-1)}, & \text{if } 1 < a \le s; \\ 1N_{s(t-1)2},\ 0N^R_{(s-1)t1}, & \text{if } 1 = a < s; \\ 1N_{1(t-1)1},\ 01^t, & \text{if } 1 = a = s. \end{cases} \tag{34}$$

Also, of course, $N_{s0a} = 0^s$.

Let us set $A_{st} = N_{st1}$ and $B_{st} = N_{st2}$. These near-perfect listings, discovered by Phillip J. Chase in 1976, have the net effect of shifting a leftmost block of 1s to the right by one or two positions, respectively, and they satisfy the following mutual recursions:

$$A_{st} = 1B_{s(t-1)},\ 0A^R_{(s-1)t}; \qquad B_{st} = 1A_{s(t-1)},\ 0A_{(s-1)t}. \tag{35}$$

"To take one step forward, take two steps forward, then one step backward; to take two steps forward, take one step forward, then another." These equations

Table 2

CHASE'S SEQUENCES FOR (3,3)-COMBINATIONS

$A_{33} = \widehat{C}_{33}^R$				$B_{33} = C_{33}$			
543	531	321	420	543	520	432	410
541	530	320	421	542	510	430	210
540	510	310	431	540	530	431	310
542	520	210	430	541	531	421	320
532	521	410	432	521	532	420	321

hold for all integer values of s and t, if we define A_{st} and B_{st} to be $\emptyset$ when s or t is negative, except that $A_{00} = B_{00} = \epsilon$ (the empty string). Thus A_{st} actually takes $\min(s,1)$ forward steps, and B_{st} actually takes $\min(s,2)$. For example, Table 2 shows the relevant listings for $s = t = 3$, using an equivalent index-list form $c_3c_2c_1$ instead of the bit strings $a_5a_4a_3a_2a_1a_0$.

Chase noticed that a computer implementation of these sequences becomes simpler if we define

$$C_{st} = \begin{cases} A_{st}, & \text{if } s+t \text{ is odd;} \\ B_{st}, & \text{if } s+t \text{ is even;} \end{cases} \qquad \widehat{C}_{st} = \begin{cases} A_{st}^R, & \text{if } s+t \text{ is even;} \\ B_{st}^R, & \text{if } s+t \text{ is odd.} \end{cases} \tag{36}$$

[See *Congressus Numerantium* **69** (1989), 215–242.] Then we have

$$C_{st} = \begin{cases} 1C_{s(t-1)},\ 0\widehat{C}_{(s-1)t}, & \text{if } s+t \text{ is odd;} \\ 1C_{s(t-1)},\ 0C_{(s-1)t}, & \text{if } s+t \text{ is even;} \end{cases} \tag{37}$$

$$\widehat{C}_{st} = \begin{cases} 0C_{(s-1)t},\ 1\widehat{C}_{s(t-1)}, & \text{if } s+t \text{ is even;} \\ 0\widehat{C}_{(s-1)t},\ 1\widehat{C}_{s(t-1)}, & \text{if } s+t \text{ is odd.} \end{cases} \tag{38}$$

When bit a_j is ready to change, we can tell where we are in the recursion by testing whether j is even or odd.

Indeed, the sequence C_{st} can be generated by a surprisingly simple algorithm, based on general ideas that apply to *any* genlex scheme. Let us say that bit a_j is *active* in a genlex algorithm if it is supposed to change before anything to its left is altered. (The node for an active bit in the corresponding trie is not the rightmost child of its parent.) Suppose we have an auxiliary table $w_n \ldots w_1w_0$, where $w_j = 1$ if and only if either a_j is active or $j < r$, where r is the least subscript such that $a_r \neq a_0$; we also let $w_n = 1$. Then the following method will find the successor of $a_{n-1} \ldots a_1a_0$:

> Set $j \leftarrow r$. If $w_j = 0$, set $w_j \leftarrow 1$, $j \leftarrow j+1$, and repeat until $w_j = 1$. Terminate if $j = n$; otherwise set $w_j \leftarrow 0$. Change a_j to $1 - a_j$, and make any other changes to $a_{j-1} \ldots a_0$ and r that apply to the particular genlex scheme being used. (39)

The beauty of this approach comes from the fact that the loop is guaranteed to be efficient: We can prove that the operation $j \leftarrow j+1$ will be performed less than once per generation step, on the average (see exercise 36).

By analyzing the transitions that occur when bits change in (37) and (38), we can readily flesh out the remaining details:

Algorithm C (*Chase's sequence*). This algorithm visits all (s,t)-combinations $a_{n-1}\ldots a_1 a_0$, where $n=s+t$, in the near-perfect order of Chase's sequence C_{st}.

C1. [Initialize.] Set $a_j \leftarrow 0$ for $0 \le j < s$, $a_j \leftarrow 1$ for $s \le j < n$, and $w_j \leftarrow 1$ for $0 \le j \le n$. If $s > 0$, set $r \leftarrow s$; otherwise set $r \leftarrow t$.

C2. [Visit.] Visit the combination $a_{n-1}\ldots a_1 a_0$.

C3. [Find j and branch.] Set $j \leftarrow r$. If $w_j = 0$, set $w_j \leftarrow 1$, $j \leftarrow j+1$, and repeat until $w_j = 1$. Terminate if $j = n$; otherwise set $w_j \leftarrow 0$ and make a four-way branch: Go to C4 if j is odd and $a_j \ne 0$, to C5 if j is even and $a_j \ne 0$, to C6 if j is even and $a_j = 0$, to C7 if j is odd and $a_j = 0$.

C4. [Move right one.] Set $a_{j-1} \leftarrow 1$, $a_j \leftarrow 0$. If $r = j > 1$, set $r \leftarrow j-1$; otherwise if $r = j-1$ set $r \leftarrow j$. Return to C2.

C5. [Move right two.] If $a_{j-2} \ne 0$, go to C4. Otherwise set $a_{j-2} \leftarrow 1$, $a_j \leftarrow 0$. If $r = j$, set $r \leftarrow \max(j-2, 1)$; otherwise if $r = j-2$, set $r \leftarrow j-1$. Return to C2.

C6. [Move left one.] Set $a_j \leftarrow 1$, $a_{j-1} \leftarrow 0$. If $r = j > 1$, set $r \leftarrow j-1$; otherwise if $r = j-1$ set $r \leftarrow j$. Return to C2.

C7. [Move left two.] If $a_{j-1} \ne 0$, go to C6. Otherwise set $a_j \leftarrow 1$, $a_{j-2} \leftarrow 0$. If $r = j-2$, set $r \leftarrow j$; otherwise if $r = j-1$, set $r \leftarrow j-2$. Return to C2. ▮

***Analysis of Chase's sequence.** The magical properties of Algorithm C cry out for further exploration, and a closer look turns out to be quite instructive. Given a bit string $a_{n-1}\ldots a_1 a_0$, let us define $a_n = 1$, $u_n = n \bmod 2$, and

$$u_j = (1-u_{j+1})a_{j+1}, \quad v_j = (u_j + j) \bmod 2, \quad w_j = (v_j + a_j) \bmod 2, \tag{40}$$

for $n > j \ge 0$. For example, we might have $n = 26$ and

$$\begin{aligned} a_{25}\ldots a_1 a_0 &= 11001001000011111101101010,\\ u_{25}\ldots u_1 u_0 &= 10100100100001010100100101,\\ v_{25}\ldots v_1 v_0 &= 00001110001011111110001111,\\ w_{25}\ldots w_1 w_0 &= 11000111001000000011100101. \end{aligned} \tag{41}$$

With these definitions we can prove by induction that $v_j = 0$ if and only if bit a_j is being "controlled" by C rather than by $\widehat{C}$ in the recursions (37)–(38) that generate $a_{n-1}\ldots a_1 a_0$, except when a_j is part of the final run of 0s or 1s at the right end. Therefore w_j agrees with the value computed by Algorithm C at the moment when $a_{n-1}\ldots a_1 a_0$ is visited, for $r \le j < n$. These formulas can be used to determine exactly where a given combination appears in Chase's sequence (see exercise 39).

If we want to work with the index-list form $c_t \ldots c_2 c_1$ instead of the bit strings $a_{n-1}\ldots a_1 a_0$, it is convenient to change the notation slightly, writing

$C_t(n)$ for C_{st} and $\widehat{C}_t(n)$ for $\widehat{C}_{st}$ when $s+t=n$. Then $C_0(n)=\widehat{C}_0(n)=\epsilon$, and the recursions for $t \ge 0$ take the form

$$C_{t+1}(n+1) = \begin{cases} nC_t(n),\ \widehat{C}_{t+1}(n), & \text{if } n \text{ is even;} \\ nC_t(n),\ C_{t+1}(n), & \text{if } n \text{ is odd;} \end{cases} \tag{42}$$

$$\widehat{C}_{t+1}(n+1) = \begin{cases} C_{t+1}(n),\ n\widehat{C}_t(n), & \text{if } n \text{ is odd;} \\ \widehat{C}_{t+1}(n),\ n\widehat{C}_t(n), & \text{if } n \text{ is even.} \end{cases} \tag{43}$$

These new equations can be expanded to tell us, for example, that

$$\begin{aligned} C_{t+1}(9) &= 8C_t(8),\ 6C_t(6),\ 4C_t(4),\ \ldots,\ 3\widehat{C}_t(3),\ 5\widehat{C}_t(5),\ 7\widehat{C}_t(7); \\ C_{t+1}(8) &= 7C_t(7),\ 6C_t(6),\ 4C_t(4),\ \ldots,\ 3\widehat{C}_t(3),\ 5\widehat{C}_t(5); \\ \widehat{C}_{t+1}(9) &= 6C_t(6),\ 4C_t(4),\ \ldots,\ 3\widehat{C}_t(3),\ 5\widehat{C}_t(5),\ 7\widehat{C}_t(7),\ 8\widehat{C}_t(8); \\ \widehat{C}_{t+1}(8) &= 6C_t(6),\ 4C_t(4),\ \ldots,\ 3\widehat{C}_t(3),\ 5\widehat{C}_t(5),\ 7\widehat{C}_t(7); \end{aligned} \tag{44}$$

notice that the same pattern predominates in all four sequences. The meaning of "..." in the middle depends on the value of t: We simply omit all terms $nC_t(n)$ and $n\widehat{C}_t(n)$ where $n < t$.

Except for edge effects at the very beginning or end, all of the expansions in (44) are based on the infinite progression

$$\ldots,\ 10,\ 8,\ 6,\ 4,\ 2,\ 0,\ 1,\ 3,\ 5,\ 7,\ 9,\ \ldots, \tag{45}$$

which is a natural way to arrange the nonnegative integers into a doubly infinite sequence. If we omit all terms of (45) that are $< t$, given any integer $t \ge 0$, the remaining terms retain the property that adjacent elements differ by either 1 or 2. Richard Stanley has suggested the name *endo-order* for this sequence, because we can remember it by thinking "even numbers decreasing, odd ...". (Notice that if we retain only the terms less than N and complement with respect to N, endo-order becomes organ-pipe order; see exercise 6.1–18.)

We could program the recursions of (42) and (43) directly, but it is interesting to unwind them using (44), thus obtaining an iterative algorithm analogous to Algorithm C. The result needs only $O(t)$ memory locations, and it is especially efficient when t is relatively small compared to n. Exercise 45 contains the details.

***Near-perfect multiset permutations.** Chase's sequences lead in a natural way to an algorithm that will generate permutations of any desired multiset $\{s_0 \cdot 0, s_1 \cdot 1, \ldots, s_d \cdot d\}$ in a near-perfect manner, meaning that

i) every transition is either $a_{j+1}a_j \leftrightarrow a_ja_{j+1}$ or $a_{j+1}a_ja_{j-1} \leftrightarrow a_{j-1}a_ja_{j+1}$;

ii) transitions of the second kind have $a_j = \min(a_{j-1}, a_{j+1})$.

Algorithm C tells us how to do this when $d=1$, and we can extend it to larger values of d by the following recursive construction [*CACM* **13** (1970), 368–369, 376]: Suppose

$$\alpha_0,\ \alpha_1,\ \ldots,\ \alpha_{N-1}$$

is any near-perfect listing of the permutations of $\{s_1\cdot 1,\ldots,s_d\cdot d\}$. Then Algorithm C, with $s=s_0$ and $t=s_1+\cdots+s_d$, tells us how to generate a listing

$$\Lambda_j \;=\; \alpha_j 0^s,\ \ldots,\ 0^a\alpha_j 0^{s-a} \qquad (46)$$

in which all transitions are $0x\leftrightarrow x0$ or $00x\leftrightarrow x00$; the final entry has $a=1$ or 2 leading zeros, depending on s and t. Therefore all transitions of the sequence

$$\Lambda_0,\ \Lambda_1^R,\ \Lambda_2,\ \ldots,\ (\Lambda_{N-1} \text{ or } \Lambda_{N-1}^R) \qquad (47)$$

are near-perfect; and this list clearly contains all the permutations.

For example, the permutations of $\{0,0,0,1,1,2\}$ generated in this way are

211000, 210100, 210001, 210010, 200110, 200101, 200011, 201001, 201010, 201100,
021100, 021001, 021010, 020110, 020101, 020011, 000211, 002011, 002101, 002110,
001120, 001102, 001012, 000112, 010012, 010102, 010120, 011020, 011002, 011200,
101200, 101020, 101002, 100012, 100102, 100120, 110020, 110002, 110200, 112000,
121000, 120100, 120001, 120010, 100210, 100201, 100021, 102001, 102010, 102100,
012100, 012001, 012010, 010210, 010201, 010021, 000121, 001021, 001201, 001210.

***Perfect schemes.** Why should we settle for a near-perfect generator like C_{st}, instead of insisting that all transitions have the simplest possible form $01\leftrightarrow 10$?

One reason is that perfect schemes don't always exist. For example, we observed in 7.2.1.2–(2) that there is no way to generate all six permutations of $\{1,1,2,2\}$ with adjacent interchanges; thus there is no perfect scheme for $(2,2)$-combinations. In fact, our chances of achieving perfection are only about 1 in 4:

Theorem P. *The generation of all (s,t)-combinations $a_{s+t-1}\ldots a_1a_0$ by adjacent interchanges $01\leftrightarrow 10$ is possible if and only if $s\le 1$ or $t\le 1$ or st is odd.*

Proof. Consider all permutations of the multiset $\{s\cdot 0, t\cdot 1\}$. We learned in exercise 5.1.2–16 that the number m_k of such permutations having k inversions is the coefficient of z^k in the z-nomial coefficient

$$\binom{s+t}{t}_z \;=\; \prod_{k=s+1}^{s+t}(1+z+\cdots+z^{k-1})\Big/\prod_{k=1}^{t}(1+z+\cdots+z^{k-1}). \qquad (48)$$

Every adjacent interchange changes the number of inversions by ± 1, so a perfect generation scheme is possible only if approximately half of all the permutations have an odd number of inversions. More precisely, the value of $\binom{s+t}{t}_{-1}=m_0-m_1+m_2-\cdots$ must be 0 or ± 1. But exercise 49 shows that

$$\binom{s+t}{t}_{-1} \;=\; \binom{\lfloor (s+t)/2\rfloor}{\lfloor t/2\rfloor}[st \text{ is even}], \qquad (49)$$

and this quantity exceeds 1 unless $s\le 1$ or $t\le 1$ or st is odd.

Conversely, perfect schemes are easy with $s\le 1$ or $t\le 1$, and they turn out to be possible also whenever st is odd. The first nontrivial case occurs for $s=t=3$, when there are four essentially different solutions; the most symmetrical of these is

210 — 310 — 410 — 510 — 520 — 521 — 531 — 532 — 432 — 431 —
421 — 321 — 320 — 420 — 430 — 530 — 540 — 541 — 542 — 543 (50)

(see exercise 51). Several authors have constructed Hamiltonian paths in the relevant graph for arbitrary odd numbers s and t; for example, the method of Eades, Hickey, and Read [*JACM* **31** (1984), 19–29] makes an interesting exercise in programming with recursive coroutines. Unfortunately, however, none of the known constructions are sufficiently simple to describe in a short space, or to implement with reasonable efficiency. Perfect combination generators have therefore not yet proved to be of practical importance. ▮

In summary, then, we have seen that the study of (s,t)-combinations leads to many fascinating patterns, some of which are of great practical importance and some of which are merely elegant and/or beautiful. Figure 26 illustrates the principal options that are available in the case $s=t=5$, when $\binom{10}{5}=252$ combinations arise. Lexicographic order (Algorithm L), the revolving-door Gray code (Algorithm R), the homogeneous scheme K_{55} of (31), and Chase's near-perfect scheme (Algorithm C) are shown in parts (a), (b), (c), and (d) of the illustration. Part (e) shows the near-perfect scheme that is as close to perfection as possible while still being in genlex order of the c array (see exercise 34), while part (f) is the perfect scheme of Eades, Hickey, and Read. Finally, Figs. 26(g) and 26(h) are listings that proceed by rotating $a_ja_{j-1}\ldots a_0 \leftarrow a_{j-1}\ldots a_0a_j$ or by swapping $a_j \leftrightarrow a_0$, akin to Algorithms 7.2.1.2C and 7.2.1.2E (see exercises 55 and 56).

***Combinations of a multiset.** If multisets can have permutations, they can have combinations too. For example, consider the multiset $\{b,b,b,b,g,g,g,r,r,r,w,w\}$, representing a sack that contains four blue balls and three that are green, three red, two white. There are 37 ways to choose five balls from this sack; in lexicographic order (but descending in each combination) they are

$$\begin{aligned}&gbbbb,\ ggbbb,\ gggbb,\ rbbbb,\ rgbbb,\ rggbb,\ rgggb,\ rrbbb,\ rrgbb,\ rrggb,\\ &rrggg,\ rrrbb,\ rrrgb,\ rrrgg,\ wbbbb,\ wgbbb,\ wggbb,\ wgggb,\ wrbbb,\ wrgbb,\\ &wrggb,\ wrggg,\ wrrbb,\ wrrgb,\ wrrgg,\ wrrrb,\ wrrrg,\ wwbbb,\ wwgbb,\ wwggb,\\ &wwggg,\ wwrbb,\ wwrgb,\ wwrgg,\ wwrrb,\ wwrrg,\ wwrrr.\end{aligned} \tag{51}$$

This fact might seem frivolous and/or esoteric, yet we will see in Theorem W below that the lexicographic generation of multiset combinations yields optimal solutions to significant combinatorial problems.

James Bernoulli observed in his *Ars Conjectandi* (1713), 119–123, that we can enumerate such combinations by looking at the coefficient of z^5 in the product $(1+z+z^2)(1+z+z^2+z^3)^2(1+z+z^2+z^3+z^4)$. Indeed, his observation is easy to understand, because we get all possible selections from the sack if we multiply out the polynomials

$$(1+w+ww)(1+r+rr+rrr)(1+g+gg+ggg)(1+b+bb+bbb+bbbb).$$

Multiset combinations are also equivalent to *bounded compositions*, namely to compositions in which the individual parts are bounded. For example, the 37 multicombinations listed in (51) correspond to 37 solutions of

$$5=r_3+r_2+r_1+r_0,\quad 0\le r_3\le 2,\quad 0\le r_2,r_1\le 3,\quad 0\le r_0\le 4,$$

namely $5=0{+}0{+}1{+}4=0{+}0{+}2{+}3=0{+}0{+}3{+}2=0{+}1{+}0{+}4=\cdots=2{+}3{+}0{+}0$.

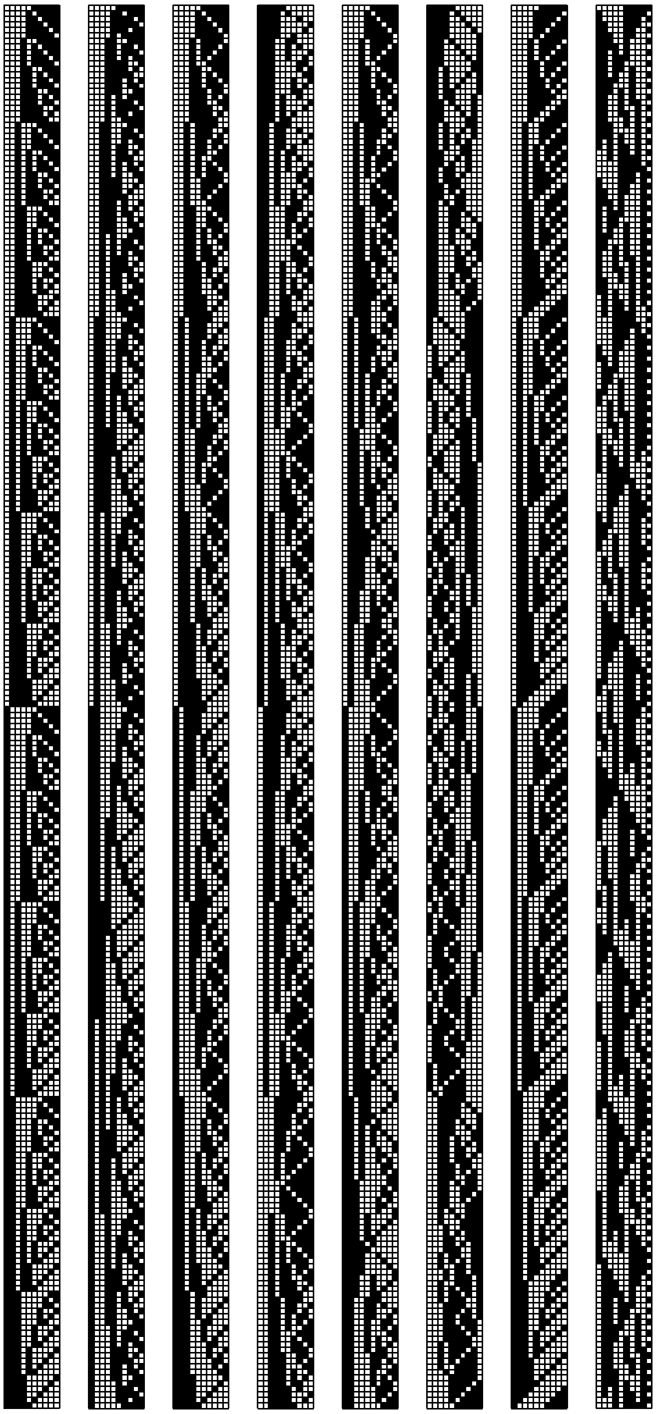


Fig. 26. Examples of $(5,5)$-combinations:
a) lexicographic;
b) revolving-door;
c) homogeneous;
d) near-perfect;
e) nearer-perfect;
f) perfect;
g) suffix-rotated;
h) right-swapped.

Bounded compositions, in turn, are special cases of *contingency tables*, which are of great importance in statistics. And all of these combinatorial configurations can be generated with Gray-like codes as well as in lexicographic order. Exercises 60–63 explore some of the basic ideas involved.

***Shadows.** Sets of combinations appear frequently in mathematics. For example, a set of 2-combinations (namely a set of pairs) is essentially a graph, and a set of t-combinations for general t is called a uniform hypergraph. If the vertices of a convex polyhedron are perturbed slightly, so that no three are collinear, no four lie in a plane, and in general no $t+1$ lie in a $(t-1)$-dimensional hyperplane, the resulting $(t-1)$-dimensional faces are "simplexes" whose vertices have great significance in computer applications. Researchers have learned that such sets of combinations have important properties related to lexicographic generation.

If α is any t-combination $c_t \ldots c_2 c_1$, its *shadow* $\partial\alpha$ is the set of all its $(t-1)$-element subsets $c_{t-1} \ldots c_2 c_1, \ldots, c_t \ldots c_3 c_1, c_t \ldots c_3 c_2$. For example, $\partial 5310 = \{310, 510, 530, 531\}$. We can also represent a t-combination as a bit string $a_{n-1} \ldots a_1 a_0$, in which case $\partial\alpha$ is the set of all strings obtained by changing a 1 to a 0: $\partial 101011 = \{001011, 100011, 101001, 101010\}$. If A is any set of t-combinations, we define its shadow

$$\partial A \;=\; \bigcup\{\, \partial\alpha \mid \alpha \in A \,\} \tag{52}$$

to be the set of all $(t-1)$-combinations in the shadows of its members. For example, $\partial\partial 5310 = \{10, 30, 31, 50, 51, 53\}$.

These definitions apply also to combinations with repetitions, namely to multicombinations: $\partial 5330 = \{330, 530, 533\}$ and $\partial\partial 5330 = \{30, 33, 50, 53\}$. In general, when A is a set of t-element multisets, ∂A is a set of $(t-1)$-element multisets. Notice, however, that ∂A never has repeated elements itself.

The *upper shadow* $\varrho\alpha$ with respect to a universe U is defined similarly, but it goes from t-combinations to $(t+1)$-combinations:

$$\varrho\alpha \;=\; \{\, \beta \subseteq U \mid \alpha \in \partial\beta \,\}, \qquad \text{for } \alpha \in U; \tag{53}$$

$$\varrho A \;=\; \bigcup\{\, \varrho\alpha \mid \alpha \in A \,\}, \qquad \text{for } A \subseteq U. \tag{54}$$

If, for example, $U = \{0, 1, 2, 3, 4, 5, 6\}$, we have $\varrho 5310 = \{53210, 54310, 65310\}$; on the other hand, if $U = \{\infty\cdot 0, \infty\cdot 1, \ldots, \infty\cdot 6\}$, we have $\varrho 5310 = \{53100, 53110, 53210, 53310, 54310, 55310, 65310\}$.

The following fundamental theorems, which have many applications in various branches of mathematics and computer science, tell us how small a set's shadows can be:

Theorem K. *If A is a set of N t-combinations contained in $U = \{0, 1, \ldots, n-1\}$, then*

$$|\partial A| \ge |\partial P_{Nt}| \qquad \textit{and} \qquad |\varrho A| \ge |\varrho Q_{Nnt}|, \tag{55}$$

where P_{Nt} denotes the first N combinations generated by Algorithm L, namely the N lexicographically smallest combinations $c_t \ldots c_2 c_1$ that satisfy (3), and Q_{Nnt} denotes the N lexicographically largest. ▮

Theorem M. *If A is a set of N t-multicombinations contained in the multiset $U = \{\infty\cdot 0, \infty\cdot 1, \ldots, \infty\cdot s\}$, then*

$$|\partial A| \ge |\partial \widehat{P}_{Nt}| \qquad \text{and} \qquad |\varrho A| \ge |\varrho \widehat{Q}_{Nst}|, \tag{56}$$

where $\widehat{P}_{Nt}$ denotes the N lexicographically smallest multicombinations $d_t \ldots d_2 d_1$ that satisfy (6), *and $\widehat{Q}_{Nst}$ denotes the N lexicographically largest.* ▮

Both of these theorems are consequences of a stronger result that we shall prove later. Theorem K is generally called the Kruskal–Katona theorem, because it was discovered by J. B. Kruskal [*Math. Optimization Techniques*, edited by R. Bellman (1963), 251–278] and rediscovered by G. Katona [*Theory of Graphs*, Tihany 1966, edited by Erdős and Katona (Academic Press, 1968), 187–207]; M. P. Schützenberger had previously stated it in a less-well-known publication, with incomplete proof [*RLE Quarterly Progress Report* **55** (1959), 117–118]. Theorem M goes back to F. S. Macaulay, many years earlier [*Proc. London Math. Soc.* (2) **26** (1927), 531–555].

Before proving (55) and (56), let's take a closer look at what those formulas mean. We know from Theorem L that the first N of all t-combinations visited by Algorithm L are those that precede $n_t \ldots n_2 n_1$, where

$$N = \binom{n_t}{t} + \cdots + \binom{n_2}{2} + \binom{n_1}{1}, \qquad n_t > \cdots > n_2 > n_1 \ge 0$$

is the degree-t combinatorial representation of N. Sometimes this representation has fewer than t nonzero terms, because n_j can be equal to $j-1$; let's suppress the zeros, and write

$$N = \binom{n_t}{t} + \binom{n_{t-1}}{t-1} + \cdots + \binom{n_v}{v}, \qquad n_t > n_{t-1} > \cdots > n_v \ge v \ge 1. \tag{57}$$

Now the first $\binom{n_t}{t}$ combinations $c_t \ldots c_1$ are the t-combinations of $\{0, \ldots, n_t - 1\}$; the next $\binom{n_{t-1}}{t-1}$ are those in which $c_t = n_t$ and $c_{t-1} \ldots c_1$ is a $(t-1)$-combination of $\{0, \ldots, n_{t-1} - 1\}$; and so on. For example, if $t = 5$ and $N = \binom{9}{5} + \binom{7}{4} + \binom{4}{3}$, the first N combinations are

$$P_{N5} = \{43210, \ldots, 87654\} \cup \{93210, \ldots, 96543\} \cup \{97210, \ldots, 97321\}. \tag{58}$$

The shadow of this set P_{N5} is, fortunately, easy to understand: It is

$$\partial P_{N5} = \{3210, \ldots, 8765\} \cup \{9210, \ldots, 9654\} \cup \{9710, \ldots, 9732\}, \tag{59}$$

namely the first $\binom{9}{4} + \binom{7}{3} + \binom{4}{2}$ combinations in lexicographic order when $t = 4$.

In other words, if we define Kruskal's function κ_t by the formula

$$\kappa_t N = \binom{n_t}{t-1} + \binom{n_{t-1}}{t-2} + \cdots + \binom{n_v}{v-1} \tag{60}$$

when N has the unique representation (57), we have

$$\partial P_{Nt} \;=\; P_{(\kappa_t N)(t-1)}. \tag{61}$$

Theorem K tells us, for example, that a graph with a million edges can contain at most

$$\binom{1414}{3} + \binom{1009}{2} = 470{,}700{,}300$$

triangles, that is, at most 470,700,300 sets of vertices $\{u, v, w\}$ with $u \mathrel{—} v \mathrel{—} w \mathrel{—} u$. The reason is that $1000000 = \binom{1414}{2} + \binom{1009}{1}$ by exercise 17, and the edges $P_{(1000000)2}$ do support $\binom{1414}{3} + \binom{1009}{2}$ triangles; but if there were more, the graph would necessarily have at least $\kappa_3 470700301 = \binom{1414}{2} + \binom{1009}{1} + \binom{1}{0} = 1000001$ edges in their shadow.

Kruskal defined the companion function

$$\lambda_t N = \binom{n_t}{t+1} + \binom{n_{t-1}}{t} + \cdots + \binom{n_v}{v+1} \tag{62}$$

to deal with questions such as this. The κ and λ functions are related by an interesting law proved in exercise 72:

$$M + N = \binom{s+t}{t} \quad \text{implies} \quad \kappa_s M + \lambda_t N = \binom{s+t}{t+1}, \quad \text{if } st > 0. \tag{63}$$

Turning to Theorem M, the sizes of $\partial \widehat{P}_{Nt}$ and $\varrho \widehat{Q}_{Nst}$ turn out to be

$$|\partial \widehat{P}_{Nt}| = \mu_t N \qquad \text{and} \qquad |\varrho \widehat{Q}_{Nst}| = N + \kappa_s N \tag{64}$$

(see exercise 81), where the function μ_t satisfies

$$\mu_t N = \binom{n_t - 1}{t-1} + \binom{n_{t-1} - 1}{t-2} + \cdots + \binom{n_v - 1}{v-1} \tag{65}$$

when N has the combinatorial representation (57).

Table 3 shows how these functions $\kappa_t N$, $\lambda_t N$, and $\mu_t N$ behave for small values of t and N. When t and N are large, they can be well approximated in terms of a remarkable function $\tau(x)$ introduced by Teiji Takagi in 1903; see Fig. 27 and exercises 82–85.

Theorems K and M are corollaries of a much more general theorem of discrete geometry, discovered by Da-Lun Wang and Ping Wang [*SIAM J. Applied Math.* **33** (1977), 55–59], which we shall now proceed to investigate. Consider the *discrete n-dimensional torus* $T(m_1, \ldots, m_n)$ whose elements are integer vectors $x = (x_1, \ldots, x_n)$ with $0 \le x_1 < m_1, \ldots, 0 \le x_n < m_n$. We define the sum and difference of two such vectors x and y as in Eqs. 4.3.2–(2) and 4.3.2–(3):

$$x + y = \big((x_1 + y_1) \bmod m_1, \ldots, (x_n + y_n) \bmod m_n\big), \tag{66}$$

$$x - y = \big((x_1 - y_1) \bmod m_1, \ldots, (x_n - y_n) \bmod m_n\big). \tag{67}$$

We also define the so-called *cross order* on such vectors by saying that $x \preceq y$ if and only if

$$\nu x < \nu y \quad \text{or} \quad (\nu x = \nu y \text{ and } x \ge y \text{ lexicographically}); \tag{68}$$

here, as usual, $\nu(x_1, \ldots, x_n) = x_1 + \cdots + x_n$. For example, when $m_1 = m_2 = 2$ and $m_3 = 3$, the 12 vectors $x_1x_2x_3$ in cross order are

$$000,\ 100,\ 010,\ 001,\ 110,\ 101,\ 011,\ 002,\ 111,\ 102,\ 012,\ 112, \tag{69}$$

Table 3

EXAMPLES OF THE KRUSKAL–MACAULAY FUNCTIONS κ, λ, AND μ

$N =$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\kappa_1 N =$	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\kappa_2 N =$	0	2	3	3	4	4	4	5	5	5	5	6	6	6	6	6	7	7	7	7	7
$\kappa_3 N =$	0	3	5	6	6	8	9	9	10	10	10	12	13	13	14	14	14	15	15	15	15
$\kappa_4 N =$	0	4	7	9	10	10	13	15	16	16	18	19	19	20	20	20	23	25	26	26	28
$\kappa_5 N =$	0	5	9	12	14	15	15	19	22	24	25	25	28	30	31	31	33	34	34	35	35
$\lambda_1 N =$	0	0	1	3	6	10	15	21	28	36	45	55	66	78	91	105	120	136	153	171	190
$\lambda_2 N =$	0	0	0	1	1	2	4	4	5	7	10	10	11	13	16	20	20	21	23	26	30
$\lambda_3 N =$	0	0	0	0	1	1	1	2	2	3	5	5	5	6	6	7	9	9	10	12	15
$\lambda_4 N =$	0	0	0	0	0	1	1	1	1	2	2	2	3	3	4	6	6	6	6	7	7
$\lambda_5 N =$	0	0	0	0	0	0	1	1	1	1	1	2	2	2	2	3	3	3	4	4	5
$\mu_1 N =$	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\mu_2 N =$	0	1	2	2	3	3	3	4	4	4	4	5	5	5	5	5	6	6	6	6	6
$\mu_3 N =$	0	1	2	3	3	4	5	5	6	6	6	7	8	8	9	9	9	10	10	10	10
$\mu_4 N =$	0	1	2	3	4	4	5	6	7	7	8	9	9	10	10	10	11	12	13	13	14
$\mu_5 N =$	0	1	2	3	4	5	5	6	7	8	9	9	10	11	12	12	13	14	14	15	15

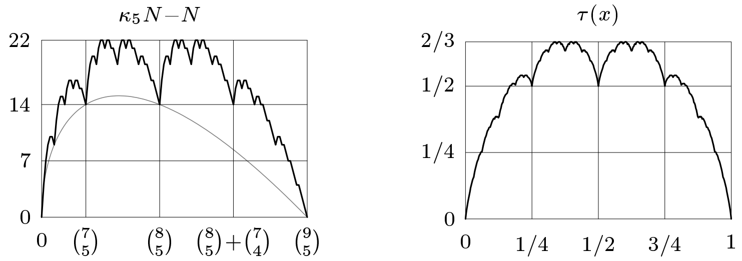


Fig. 27. Approximating a Kruskal function with the Takagi function. (The smooth curve in the left-hand graph is the lower bound $\underline{\kappa}_5 N - N$ of exercise 80.)

omitting parentheses and commas for convenience. The *complement* of a vector in $T(m_1, \ldots, m_n)$ is

$$\overline{x} \;=\; (m_1 - 1 - x_1, \ldots, m_n - 1 - x_n). \tag{70}$$

Notice that $x \preceq y$ holds if and only if $\overline{x} \succeq \overline{y}$. Therefore we have

$$\operatorname{rank}(x) + \operatorname{rank}(\overline{x}) \;=\; T - 1, \qquad \text{where } T = m_1 \ldots m_n, \tag{71}$$

if rank(x) denotes the number of vectors that precede x in cross order.

We will find it convenient to call the vectors "points" and to name the points $e_0, e_1, \ldots, e_{T-1}$ in increasing cross order. Thus we have $e_7 = 002$ in (69), and $\overline{e}_r = e_{T-1-r}$ in general. Notice that

$$e_1 = 100\ldots00, \quad e_2 = 010\ldots00, \quad \ldots, \quad e_n = 000\ldots01; \tag{72}$$

these are the so-called *unit vectors*. The set

$$S_N \;=\; \{e_0, e_1, \ldots, e_{N-1}\} \tag{73}$$

consisting of the smallest N points is called a *standard set*, and in the special case $N = n + 1$ we write

$$E = \{e_0, e_1, \ldots, e_n\} = \{000\ldots00, 100\ldots00, 010\ldots00, \ldots, 000\ldots01\}. \tag{74}$$

Any set of points X has a *spread* X^+, a *core* X°, and a *dual* $X^\sim$, defined by the rules

$$X^+ \;=\; \{\, x \in S_T \mid x \in X \text{ or } x - e_1 \in X \text{ or } \cdots \text{ or } x - e_n \in X \,\}; \tag{75}$$
$$X^\circ \;=\; \{\, x \in S_T \mid x \in X \text{ and } x + e_1 \in X \text{ and } \cdots \text{ and } x + e_n \in X \,\}; \tag{76}$$
$$X^\sim \;=\; \{\, x \in S_T \mid \overline{x} \notin X \,\}. \tag{77}$$

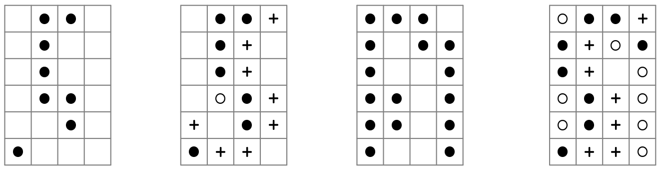
We can also define the spread of X algebraically, writing

$$X^+ \;=\; X + E, \tag{78}$$

where $X + Y$ denotes $\{\, x + y \mid x \in X \text{ and } y \in Y \,\}$. Clearly

$$X^+ \subseteq Y \qquad \text{if and only if} \qquad X \subseteq Y^\circ. \tag{79}$$

These notions can be illustrated in the two-dimensional case $m_1 = 4$, $m_2 = 6$, by the more-or-less random toroidal arrangement $X = \{00, 12, 13, 14, 15, 21, 22, 25\}$ for which we have, pictorially,

 ; (80)

X $\qquad$ X° and X^+ $\qquad$ $X^\sim$ $\qquad$ $X^{\sim\circ}$ and $X^{\sim+}$

here X in the first two diagrams consists of points marked $\bullet$ or $\circ$, X° comprises just the $\circ$s, and X^+ consists of +s plus $\bullet$s plus $\circ$s. Notice that if we rotate the diagram for $X^{\sim\circ}$ and $X^{\sim+}$ by 180°, we obtain the diagram for X° and X^+, but with ($\bullet, \circ, +,\ $) respectively changed to ($+,\ , \bullet, \circ$); and in fact the identities

$$X^\circ \;=\; X^{\sim+\sim}, \qquad X^+ \;=\; X^{\sim\circ\sim} \tag{81}$$

hold in general (see exercise 86).

Now we are ready to state the theorem of Wang and Wang:

Theorem W. *Let X be any set of N points in the discrete torus $T(m_1, \ldots, m_n)$, where $m_1 \le \cdots \le m_n$. Then $|X^+| \ge |S_N^+|$ and $|X^\circ| \le |S_N^\circ|$.*

In other words, the standard sets S_N have the smallest spread and largest core, among all N-point sets. We will prove this result by following a general approach first used by F. W. J. Whipple to prove Theorem M [*Proc. London Math. Soc.* (2) **28** (1928), 431–437]. The first step is to prove that the spread and the core of standard sets are standard:

Lemma S. *There are functions α and β such that $S_N^+ = S_{\alpha N}$ and $S_N^\circ = S_{\beta N}$.*

Proof. We may assume that $N > 0$. Let r be maximum with $e_r \in S_N^+$, and let $\alpha N = r + 1$; we must prove that $e_q \in S_N^+$ for $0 \le q < r$. Suppose $e_q = x = (x_1, \ldots, x_n)$ and $e_r = y = (y_1, \ldots, y_n)$, and let k be the largest subscript with $x_k > 0$. Since $y \in S_N^+$, there is a subscript j such that $y - e_j \in S_N$. It suffices to prove that $x - e_k \preceq y - e_j$, and exercise 88 does this.

The second part follows from (81), with $\beta N = T - \alpha(T - N)$, because $S_N^\sim = S_{T-N}$. ▮

Theorem W is obviously true when $n = 1$, so we assume by induction that it has been proved in $n - 1$ dimensions. The next step is to *compress* the given set X in the kth coordinate position, by partitioning it into disjoint sets

$$X_k(a) \;=\; \{\, x \in X \mid x_k = a \,\} \tag{82}$$

for $0 \le a < m_k$ and replacing each $X_k(a)$ by

$$X'_k(a) \;=\; \{\, (s_1, \ldots, s_{k-1}, a, s_k, \ldots, s_{n-1}) \mid (s_1, \ldots, s_{n-1}) \in S_{|X_k(a)|} \,\}, \tag{83}$$

a set with the same number of elements. The sets S used in (83) are standard in the $(n-1)$-dimensional torus $T(m_1, \ldots, m_{k-1}, m_{k+1}, \ldots, m_n)$. Notice that we have $(x_1, \ldots, x_{k-1}, a, x_{k+1}, \ldots, x_n) \preceq (y_1, \ldots, y_{k-1}, a, y_{k+1}, \ldots, y_n)$ if and only if $(x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n) \preceq (y_1, \ldots, y_{k-1}, y_{k+1}, \ldots, y_n)$; therefore $X'_k(a) = X_k(a)$ if and only if the $(n-1)$-dimensional points $(x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n)$ with $(x_1, \ldots, x_{k-1}, a, x_{k+1}, \ldots, x_n) \in X$ are as small as possible when projected onto the $(n-1)$-dimensional torus. We let

$$C_k X \;=\; X'_k(0) \cup X'_k(1) \cup \cdots \cup X'_k(m_k - 1) \tag{84}$$

be the compression of X in position k. Exercise 90 proves the basic fact that compression does not increase the size of the spread:

$$|X^+| \;\ge\; |(C_k X)^+|, \qquad \text{for } 1 \le k \le n. \tag{85}$$

Furthermore, if compression changes X, it replaces some of the elements by other elements of lower rank. Therefore we need to prove Theorem W only for sets X that are totally compressed, having $X = C_k X$ for all k.

Consider, for example, the case $n = 2$. A totally compressed set in two dimensions has all points moved to the left of their rows and the bottom of their columns, as in the eleven-point sets

●	+		
●	+		
●	+		
●	+	+	
●	●	●	+
●	●	●	●

or

+			
●	+		
●	+		
●	●	+	
●	●	●	+
●	●	●	●

or

+	+		
●	●	+	
●	●	+	
●	●	●	+
●	●	●	●

or

+			
●	+	+	
●	●	●	+
●	●	●	+
●	●	●	●

or

+			
●	+		
●	●	+	+
●	●	●	●
●	●	●	●

;

the rightmost of these is standard, and has the smallest spread. Exercise 91 completes the proof of Theorem W in two dimensions.

When $n > 2$, suppose $x = (x_1, \ldots, x_n) \in X$ and $x_j > 0$. The condition $C_k X = X$ implies that, if $0 \le i < j$ and $i \ne k \ne j$, we have $x + e_i - e_j \in X$. Applying this fact for three values of k tells us that $x + e_i - e_j \in X$ whenever $0 \le i < j$. Consequently

$$X_n(a) + E_n(0) \;\subseteq\; X_n(a-1) + e_n \quad \text{for } 0 < a < m, \tag{86}$$

where $m = m_n$ and $E_n(0)$ is a clever abbreviation for the set $\{e_0, \ldots, e_{n-1}\}$.

Let $X_n(a)$ have N_a elements, so that $N = |X| = N_0 + N_1 + \cdots + N_{m-1}$, and let $Y = X^+$. Then

$$Y_n(a) \;=\; \bigl(X_n\bigl((a-1) \bmod m\bigr) + e_n\bigr) \cup \bigl(X_n(a) + E_n(0)\bigr)$$

is standard in $n-1$ dimensions, and (86) tells us that

$$N_{m-1} \le \beta N_{m-2} \le N_{m-2} \le \cdots \le N_1 \le \beta N_0 \le N_0 \le \alpha N_0,$$

where α and β refer to coordinates 1 through $n-1$. Therefore

$$\begin{aligned} |Y| &= |Y_n(0)| + |Y_n(1)| + |Y_n(2)| + \cdots + |Y_n(m-1)| \\ &= \alpha N_0 + N_0 + N_1 + \cdots + N_{m-2} = \alpha N_0 + N - N_{m-1}. \end{aligned}$$

The proof of Theorem W now has a beautiful conclusion. Let $Z = S_N$, and suppose $|Z_n(a)| = M_a$. We want to prove that $|X^+| \ge |Z^+|$, namely that

$$\alpha N_0 + N - N_{m-1} \;\ge\; \alpha M_0 + N - M_{m-1}, \tag{87}$$

because the arguments of the previous paragraph apply to Z as well as to X. We will prove (87) by showing that $N_{m-1} \le M_{m-1}$ and $N_0 \ge M_0$.

Using the $(n-1)$-dimensional α and β functions, let us define

$$N'_{m-1} = N_{m-1},\; N'_{m-2} = \alpha N'_{m-1},\; \ldots,\; N'_1 = \alpha N'_2,\; N'_0 = \alpha N'_1; \tag{88}$$

$$N''_0 = N_0,\; N''_1 = \beta N''_0,\; N''_2 = \beta N''_1,\; \ldots,\; N''_{m-1} = \beta N''_{m-2}. \tag{89}$$

Then we have $N'_a \le N_a \le N''_a$ for $0 \le a < m$, and it follows that

$$N' = N'_0 + N'_1 + \cdots + N'_{m-1} \;\le\; N \;\le\; N'' = N''_0 + N''_1 + \cdots + N''_{m-1}. \tag{90}$$

Exercise 92 proves that the standard set $Z' = S_{N'}$ has exactly N'_a elements with nth coordinate equal to a, for each a; and by the duality between α and β, the standard set $Z'' = S_{N''}$ likewise has exactly N''_a elements with nth coordinate a. Finally, therefore,

$$\begin{aligned} M_{m-1} = |Z_n(m-1)| &\ge |Z'_n(m-1)| = N_{m-1}, \\ M_0 = |Z_n(0)| &\le |Z''_n(0)| = N_0, \end{aligned}$$

because $Z' \subseteq Z \subseteq Z''$ by (90). By (81) we also have $|X^\circ| \le |Z^\circ|$. ▮

Now we are ready to prove Theorems K and M, which are in fact special cases of a substantially more general theorem of Clements and Lindström that applies to arbitrary multisets [*J. Combinatorial Theory* **7** (1969), 230–238]:

Corollary C. *If A is a set of N t-multicombinations contained in the multiset $U = \{s_0 \cdot 0, s_1 \cdot 1, \ldots, s_d \cdot d\}$, where $s_0 \ge s_1 \ge \cdots \ge s_d$, then*

$$|\partial A| \ge |\partial P_{Nt}| \qquad \text{and} \qquad |\varrho A| \ge |\varrho Q_{Nt}|, \tag{91}$$

where P_{Nt} denotes the N lexicographically smallest multicombinations $d_t \ldots d_2 d_1$ of U, and Q_{Nt} denotes the N lexicographically largest.

Proof. Multicombinations of U can be represented as points $x_1 \ldots x_n$ of the torus $T(m_1, \ldots, m_n)$, where $n = d+1$ and $m_j = s_{n-j} + 1$; we let x_j be the number of occurrences of $n - j$. This correspondence preserves lexicographic order. For example, if $U = \{0,0,0,1,1,2,3\}$, its 3-multicombinations are

$$000,\ 100,\ 110,\ 200,\ 210,\ 211,\ 300,\ 310,\ 311,\ 320,\ 321, \tag{92}$$

in lexicographic order, and the corresponding points $x_1x_2x_3x_4$ are

$$0003, 0012, 0021, 0102, 0111, 0120, 1002, 1011, 1020, 1101, 1110. \tag{93}$$

Let T_w be the points of the torus that have weight $x_1 + \cdots + x_n = w$. Then every allowable set A of t-multicombinations is a subset of T_t. Furthermore — and this is the main point — the spread of $T_0 \cup T_1 \cup \cdots \cup T_{t-1} \cup A$ is

$$\begin{aligned}(T_0 \cup T_1 \cup \cdots \cup T_{t-1} \cup A)^+ &= T_0^+ \cup T_1^+ \cup \cdots \cup T_{t-1}^+ \cup A^+ \\ &= T_0 \cup T_1 \cup \cdots \cup T_t \cup \varrho A.\end{aligned} \tag{94}$$

Thus the upper shadow ϱA is simply $(T_0 \cup T_1 \cup \cdots \cup T_{t-1} \cup A)^+ \cap T_{t+1}$, and Theorem W tells us in essence that $|A| = N$ implies $|\varrho A| \ge |\varrho(S_{M+N} \cap T_t)|$, where $M = |T_0 \cup \cdots \cup T_{t-1}|$. Hence, by the definition of cross order, $S_{M+N} \cap T_t$ consists of the lexicographically largest N t-multicombinations, namely Q_{Nt}.

The proof that $|\partial A| \ge |\partial P_{Nt}|$ now follows by complementation (see exercise 94). ∎

EXERCISES

1. [*M23*] Explain why Golomb's rule (8) makes all sets $\{c_1, \ldots, c_t\} \subseteq \{0, \ldots, n-1\}$ correspond uniquely to multisets $\{e_1, \ldots, e_t\} \subseteq \{\infty \cdot 0, \ldots, \infty \cdot n-t\}$.

2. [*16*] What path in an 11×13 grid corresponds to the bit string (13)?

▸ **3.** [*21*] (R. R. Fenichel, 1968.) Show that the compositions $q_t + \cdots + q_1 + q_0$ of s into $t+1$ nonnegative parts can be generated in lexicographic order by a simple loopless algorithm.

4. [*16*] Show that every composition $q_t \ldots q_0$ of s into $t+1$ nonnegative parts corresponds to a composition $r_s \ldots r_0$ of t into $s+1$ nonnegative parts. What composition corresponds to 10224000001010 under this correspondence?

▸ **5.** [*20*] What is a good way to generate all of the integer solutions to the following systems of inequalities?

a) $n > x_t \ge x_{t-1} > x_{t-2} \ge x_{t-3} > \cdots > x_1 \ge 0$, when t is odd.

b) $n \gg x_t \gg x_{t-1} \gg \cdots \gg x_2 \gg x_1 \gg 0$, where $a \gg b$ means $a \ge b+2$.

6. [*M22*] How often is each step of Algorithm T performed?

7. [*22*] Design an algorithm that runs through the "dual" combinations $b_s \ldots b_2 b_1$ in *decreasing* lexicographic order (see (5) and Table 1). Like Algorithm T, your algorithm should avoid redundant assignments and unnecessary searching.

8. [*M23*] Design an algorithm that generates all (s,t)-combinations $a_{n-1} \ldots a_1 a_0$ lexicographically in bitstring form. The total running time should be $O(\binom{n}{t})$, assuming that $st > 0$.

9. [*M26*] When all (s,t)-combinations $a_{n-1} \ldots a_1 a_0$ are listed in lexicographic order, let $2A_{st}$ be the total number of bit changes between adjacent strings. For example, $A_{33} = 25$ because there are respectively

$$2+2+2+4+2+2+4+2+2+6+2+2+4+2+2+4+2+2+2 = 50$$

bit changes between the 20 strings in Table 1.

a) Show that $A_{st} = \min(s,t) + A_{(s-1)t} + A_{s(t-1)}$ when $st > 0$; $A_{st} = 0$ when $st = 0$.
b) Prove that $A_{st} < 2\binom{s+t}{t}$.

▸ **10.** [*21*] The "World Series" of baseball is traditionally a competition in which the American League champion (A) plays the National League champion (N) until one of them has beaten the other four times. What is a good way to list all possible scenarios AAAA, AAANA, AAANNA, ..., NNNN? What is a simple way to assign consecutive integers to those scenarios?

11. [*19*] Which of the scenarios in exercise 10 occurred most often during the 1900s? Which of them never occurred? [*Hint:* World Series scores are easily found on the Internet.]

12. [*HM32*] A set V of n-bit vectors that is closed under addition modulo 2 is called a *binary vector space.*

a) Prove that every such V contains 2^t elements, for some integer t, and can be represented as the set $\{x_1\alpha_1 \oplus \cdots \oplus x_t\alpha_t \mid 0 \le x_1, \ldots, x_t \le 1\}$ where the vectors $\alpha_1, \ldots, \alpha_t$ form a "canonical basis" with the following property: There is a t-combination $c_t \ldots c_2 c_1$ of $\{0, 1, \ldots, n-1\}$ such that, if α_k is the binary vector $a_{k(n-1)} \ldots a_{k1} a_{k0}$, we have

$$a_{kc_j} = [j = k] \quad \text{for } 1 \le j, k \le t; \qquad a_{kl} = 0 \quad \text{for } 0 \le l < c_k,\ 1 \le k \le t.$$

For example, the canonical bases with $n = 9$, $t = 4$, and $c_4c_3c_2c_1 = 7641$ have the general form

$$\begin{aligned} \alpha_1 &= *\,0\,0\,*\,0\,*\,*\,1\,0, \\ \alpha_2 &= *\,0\,0\,*\,1\,0\,0\,0\,0, \\ \alpha_3 &= *\,0\,1\,0\,0\,0\,0\,0\,0, \\ \alpha_4 &= *\,1\,0\,0\,0\,0\,0\,0\,0; \end{aligned}$$

there are 2^8 ways to replace the eight asterisks by 0s and/or 1s, and each of these defines a canonical basis. We call t the dimension of V.

b) How many t-dimensional spaces are possible with n-bit vectors?
c) Design an algorithm to generate all canonical bases $(\alpha_1, \ldots, \alpha_t)$ of dimension t. *Hint:* Let the associated combinations $c_t \ldots c_1$ increase lexicographically as in Algorithm L.
d) What is the 1000000th basis visited by your algorithm when $n = 9$ and $t = 4$?

13. [*25*] A one-dimensional *Ising configuration* of length n, weight t, and energy r, is a binary string $a_{n-1} \ldots a_0$ such that $\sum_{j=0}^{n-1} a_j = t$ and $\sum_{j=1}^{n-1} b_j = r$, where $b_j =$

$a_j \oplus a_{j-1}$. For example, $a_{12} \ldots a_0 = 1100100100011$ has weight 6 and energy 6, since $b_{12} \ldots b_1 = 010110110010$.

Design an algorithm to generate all such configurations, given n, t, and r.

14. [*26*] When the binary strings $a_{n-1} \ldots a_1 a_0$ of (s,t)-combinations are generated in lexicographic order, we sometimes need to change $2\min(s,t)$ bits to get from one combination to the next. For example, 011100 is followed by 100011 in Table 1. Therefore we apparently cannot hope to generate all combinations with a loopless algorithm unless we visit them in some other order.

Show, however, that there actually is a way to compute the lexicographic successor of a given combination in $O(1)$ steps, if each combination is represented indirectly in a doubly linked list as follows: There are arrays $l[0], \ldots, l[n]$ and $r[0], \ldots, r[n]$ such that $l[r[j]] = j$ for $0 \le j \le n$. If $x_0 = l[0]$ and $x_j = l[x_{j-1}]$ for $0 < j < n$, then $a_j = [x_j > s]$ for $0 \le j < n$.

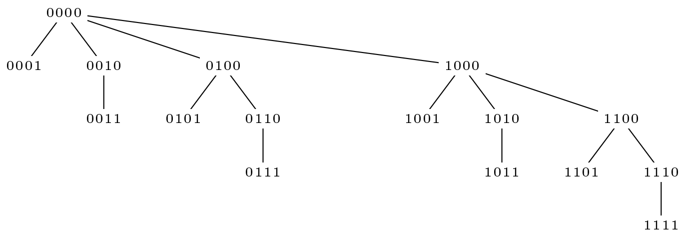
15. [*M22*] Use the fact that dual combinations $b_s \ldots b_2 b_1$ occur in reverse lexicographic order to prove that the sum $\binom{b_s}{s} + \cdots + \binom{b_2}{2} + \binom{b_1}{1}$ has a simple relation to the sum $\binom{c_t}{t} + \cdots + \binom{c_2}{2} + \binom{c_1}{1}$.

16. [*M21*] What is the millionth combination generated by Algorithm L when t is (a) 2? (b) 3? (c) 4? (d) 5? (e) 1000000?

17. [*HM25*] Given N and t, what is a good way to compute the combinatorial representation (20)?

▶ **18.** [*20*] What binary tree do we get when the binomial tree T_n is represented by "right child" and "left sibling" pointers as in exercise 2.3.2–5?

19. [*21*] Instead of labeling the branches of the binomial tree T_4 as shown in (22), we could label each node with the bit string of its corresponding combination:



If T_∞ has been labeled in this way, suppressing leading zeros, preorder is the same as the ordinary increasing order of binary notation; so the millionth node turns out to be 11110100001000111111. But what is the millionth node of T_∞ in *postorder*?

20. [*M20*] Find generating functions g and h such that Algorithm F finds exactly $[z^N]\,g(z)$ feasible combinations and sets $t \leftarrow t+1$ exactly $[z^N]\,h(z)$ times.

21. [*M22*] Prove the alternating combination law (30).

22. [*M23*] What is the millionth revolving-door combination visited by Algorithm R when t is (a) 2? (b) 3? (c) 4? (d) 5? (e) 1000000?

23. [*M23*] Suppose we augment Algorithm R by setting $j \leftarrow t+1$ in step R1, and $j \leftarrow 1$ if R3 goes directly to R2. Find the probability distribution of j, and its average value. What does this imply about the running time of the algorithm?

▶ **24.** [*M25*] (W. H. Payne, 1974.) Continuing the previous exercise, let j_k be the value of j on the kth visit by Algorithm R. Show that $|j_{k+1} - j_k| \le 2$, and explain how to make the algorithm loopless by exploiting this property.

25. [*M35*] Let $c_t \ldots c_2 c_1$ and $c'_t \ldots c'_2 c'_1$ be the Nth and N'th combinations generated by the revolving-door method, Algorithm R. If the set $C = \{c_t, \ldots, c_2, c_1\}$ has m elements not in $C' = \{c'_t, \ldots, c'_2, c'_1\}$, prove that $|N - N'| > \sum_{k=1}^{m-1} \binom{2k}{k-1}$.

26. [*26*] Do elements of the *ternary* reflected Gray code have properties similar to the revolving-door Gray code Γ_{st}, if we extract only the n-tuples $a_{n-1} \ldots a_1 a_0$ such that (a) $a_{n-1} + \cdots + a_1 + a_0 = t$? (b) $\{a_{n-1}, \ldots, a_1, a_0\} = \{r \cdot 0, s \cdot 1, t \cdot 2\}$?

▶ **27.** [*25*] Show that there is a simple way to generate all combinations of *at most* t elements of $\{0, 1, \ldots, n-1\}$, using only Gray-code-like transitions $0 \leftrightarrow 1$ and $01 \leftrightarrow 10$. (In other words, each step should either insert a new element, delete an element, or shift an element by ± 1.) For example,

$$0000,\ 0001,\ 0011,\ 0010,\ 0110,\ 0101,\ 0100,\ 1100,\ 1010,\ 1001,\ 1000$$

is one such sequence when $n = 4$ and $t = 2$. *Hint:* Think of Chinese rings.

28. [*M21*] True or false: A listing of (s,t)-combinations $a_{n-1} \ldots a_1 a_0$ in bitstring form is in genlex order if and only if the corresponding index-form listings $b_s \ldots b_2 b_1$ (for the 0s) and $c_t \ldots c_2 c_1$ (for the 1s) are both in genlex order.

▶ **29.** [*M28*] (P. J. Chase.) Given a string on the symbols `+`, `-`, and `0`, say that an *R-block* is a substring of the form $\mathtt{-}^{k+1}$ that is preceded by `0` and not followed by `-`; an *L-block* is a substring of the form $\mathtt{+-}^{k}$ that is followed by `0`; in both cases $k \ge 0$. For example, the string $\boxed{\mathtt{+}}\mathtt{00++-++}\boxed{\mathtt{+-}}\mathtt{000}\boxed{\mathtt{-}}$ has two L-blocks and one R-block, shown in gray. Notice that blocks cannot overlap.

We form the *successor* of such a string as follows, whenever at least one block is present: Replace the rightmost $\mathtt{0-}^{k+1}$ by $\mathtt{-+}^{k}\mathtt{0}$, if the rightmost block is an R-block; otherwise replace the rightmost $\mathtt{+-}^{k}\mathtt{0}$ by $\mathtt{0+}^{k+1}$. Also negate the first sign, if any, that appears to the right of the block that has been changed. For example,

$$\mathtt{-}\boxed{\mathtt{+}}\mathtt{00++-} \to \mathtt{-0}\boxed{\mathtt{+}}\mathtt{0}\boxed{\mathtt{-}}\mathtt{+-} \to \mathtt{-0}\boxed{\mathtt{+-}}\mathtt{0}\boxed{\mathtt{--}} \to \mathtt{-0+--}\boxed{\mathtt{+}}\mathtt{0} \to \mathtt{-0}\boxed{\mathtt{+--}}\mathtt{0+} \to \mathtt{-00+++-},$$

where the notation $\alpha \to \beta$ means that β is the successor of α.

a) What strings have no blocks (and therefore no successor)?
b) Can there be a cycle of strings with $\alpha_0 \to \alpha_1 \to \cdots \to \alpha_{k-1} \to \alpha_0$?
c) Prove that if $\alpha \to \beta$ then $-\beta \to -\alpha$, where "$-$" means "negate all the signs." (Therefore every string has at most one predecessor.)
d) Show that if $\alpha_0 \to \alpha_1 \to \cdots \to \alpha_k$ and $k > 0$, the strings α_0 and α_k do not have all their `0`s in the same positions. (Therefore, if α_0 has s signs and t zeros, k must be less than $\binom{s+t}{t}$.)
e) Prove that every string α with s signs and t zeros belongs to exactly one chain $\alpha_0 \to \alpha_1 \to \cdots \to \alpha_{\binom{s+t}{t}-1}$.

30. [*M32*] The previous exercise defines 2^s ways to generate all combinations of s 0s and t 1s, via the mapping $\mathtt{+} \mapsto 0$, $\mathtt{-} \mapsto 0$, and $\mathtt{0} \mapsto 1$. Show that each of these ways is a homogeneous genlex sequence, definable by an appropriate recurrence. Is Chase's sequence (37) a special case of this general construction?

31. [*M23*] How many genlex listings of (s,t)-combinations are possible in (a) bitstring form $a_{n-1} \ldots a_1 a_0$? (b) index-list form $c_t \ldots c_2 c_1$?

▸ **32.** [*M32*] How many of the genlex listings of (s,t)-combination strings $a_{n-1}\ldots a_1a_0$ (a) have the revolving-door property? (b) are homogeneous?

33. [*HM33*] How many of the genlex listings in exercise 31(b) are near-perfect?

34. [*M32*] Continuing exercise 33, explain how to find such schemes that are as near as possible to perfection, in the sense that the number of "imperfect" transitions $c_j \leftarrow c_j \pm 2$ is minimized, when s and t are not too large.

35. [*M26*] How many steps of Chase's sequence C_{st} use an imperfect transition?

▸ **36.** [*M21*] Prove that method (39) performs the operation $j \leftarrow j+1$ a total of exactly $\binom{s+t}{t}-1$ times as it generates all (s,t)-combinations $a_{n-1}\ldots a_1a_0$, given any genlex scheme for combinations in bitstring form.

▸ **37.** [*27*] What algorithm results when the general genlex method (39) is used to produce (s,t)-combinations $a_{n-1}\ldots a_1a_0$ in (a) lexicographic order? (b) the revolving-door order of Algorithm R? (c) the homogeneous order of (31)?

38. [*26*] Design a genlex algorithm like Algorithm C for the *reverse* sequence C_{st}^R.

39. [*M21*] When $s=12$ and $t=14$, how many combinations precede the bit string 11001001000011111101101010 in Chase's sequence C_{st}? (See (41).)

40. [*M22*] What is the millionth combination in Chase's sequence C_{st}, when $s=12$ and $t=14$?

41. [*M27*] Show that there is a permutation $c(0)$, $c(1)$, $c(2)$, $\ldots$ of the nonnegative integers such that the elements of Chase's sequence C_{st} are obtained by complementing the least significant $s+t$ bits of the elements $c(k)$ for $0 \le k < 2^{s+t}$ that have weight $\nu(c(k))=s$. (Thus the sequence $\bar{c}(0)$, $\ldots$, $\bar{c}(2^n-1)$ contains, as subsequences, all of the C_{st} for which $s+t=n$, just as Gray binary code $g(0)$, $\ldots$, $g(2^n-1)$ contains all the revolving-door sequences Γ_{st}.) Explain how to compute the binary representation $c(k)=(\ldots a_2a_1a_0)_2$ from the binary representation $k=(\ldots b_2b_1b_0)_2$.

42. [*HM34*] Use generating functions of the form $\sum_{s,t} g_{st}w^sz^t$ to analyze each step of Algorithm C.

43. [*20*] Prove or disprove: If $s(x)$ and $p(x)$ denote respectively the successor and predecessor of x in endo-order, then $s(x+1)=p(x)+1$.

▸ **44.** [*M21*] Let $C_t(n)-1$ denote the sequence obtained from $C_t(n)$ by striking out all combinations with $c_1=0$, then replacing $c_t\ldots c_1$ by $(c_t-1)\ldots(c_1-1)$ in the combinations that remain. Show that $C_t(n)-1$ is near-perfect.

45. [*32*] Exploit endo-order and the expansions sketched in (44) to generate the combinations $c_t\ldots c_2c_1$ of Chase's sequence $C_t(n)$ with a nonrecursive procedure.

▸ **46.** [*33*] Construct a nonrecursive algorithm for the dual combinations $b_s\ldots b_2b_1$ of Chase's sequence C_{st}, namely for the positions of the zeros in $a_{n-1}\ldots a_1a_0$.

47. [*26*] Implement the near-perfect multiset permutation method of (46) and (47).

48. [*M21*] Suppose $\alpha_0, \alpha_1, \ldots, \alpha_{N-1}$ is any listing of the permutations of the multiset $\{s_1\cdot 1,\ldots,s_d\cdot d\}$, where α_k differs from α_{k+1} by the interchange of two elements. Let β_0, $\ldots$, β_{M-1} be any revolving-door listing for (s,t)-combinations, where $s=s_0$, $t=s_1+\cdots+s_d$, and $M=\binom{s+t}{t}$. Then let Λ_j be the list of M elements obtained by starting with $\alpha_j \uparrow \beta_0$ and applying the revolving-door exchanges; here $\alpha \uparrow \beta$ denotes the string obtained by substituting the elements of α for the 1s in β, preserving left-right order. For example, if β_0, $\ldots$, β_{M-1} is 0110, 0101, 1100, 1001, 0011, 1010, and if $\alpha_j=12$, then Λ_j is 0120, 0102, 1200, 1002, 0012, 1020. (The revolving-door listing need *not* be homogeneous.)

Prove that the list (47) contains all permutations of $\{s_0 \cdot 0, s_1 \cdot 1, \ldots, s_d \cdot d\}$, and that adjacent permutations differ from each other by the interchange of two elements.

49. [*HM23*] If q is a primitive mth root of unity, such as $e^{2\pi i/m}$, show that

$$\binom{n}{k}_q = \binom{\lfloor n/m \rfloor}{\lfloor k/m \rfloor}\binom{n \bmod m}{k \bmod m}_q.$$

▸ **50.** [*HM25*] Extend the formula of the previous exercise to *q-multinomial* coefficients

$$\binom{n_1 + \cdots + n_t}{n_1, \ldots, n_t}_q.$$

51. [*25*] Find all Hamiltonian paths in the graph whose vertices are permutations of $\{0,0,0,1,1,1\}$ related by adjacent transposition. Which of those paths are equivalent under the operations of interchanging 0s with 1s and/or left-right reflection?

52. [*M37*] Generalizing Theorem P, find a necessary and sufficient condition that all permutations of the multiset $\{s_0 \cdot 0, \ldots, s_d \cdot d\}$ can be generated by adjacent transpositions $a_j a_{j-1} \leftrightarrow a_{j-1} a_j$.

53. [*M46*] (D. H. Lehmer, 1965.) Suppose the N permutations of $\{s_0 \cdot 0, \ldots, s_d \cdot d\}$ cannot be generated by a perfect scheme, because $(N + x)/2$ of them have an even number of inversions, where $x \geq 2$. Is it possible to generate them all with a sequence of $N + x - 2$ adjacent interchanges $a_{\delta_k} \leftrightarrow a_{\delta_k - 1}$ for $1 \leq k < N + x - 1$, where $x - 1$ cases are "spurs" with $\delta_k = \delta_{k-1}$ that take us back to the permutation we've just seen? For example, a suitable sequence $\delta_1 \ldots \delta_{94}$ for the 90 permutations of $\{0,0,1,1,2,2\}$, where $x = \binom{2+2+2}{2,2,2}_{-1} = 6$, is $234535432523451\alpha42\alpha^R51\alpha42\alpha^R51\alpha4$, where $\alpha = 45352542345355$, if we start with $a_5a_4a_3a_2a_1a_0 = 221100$.

54. [*M40*] For what values of s and t can all (s,t)-combinations be generated if we allow end-around swaps $a_{n-1} \leftrightarrow a_0$ in addition to adjacent interchanges $a_j \leftrightarrow a_{j-1}$?

▸ **55.** [*30*] (Frank Ruskey, 2004.) Show that all (s,t)-combinations $a_{s+t-1} \ldots a_1 a_0$ can be generated efficiently by doing successive rotations $a_j a_{j-1} \ldots a_0 \leftarrow a_{j-1} \ldots a_0 a_j$.

56. [*M49*] (Buck and Wiedemann, 1984.) Can all (t,t)-combinations $a_{2t-1} \ldots a_1 a_0$ be generated by repeatedly swapping a_0 with some other element?

▸ **57.** [*22*] (Frank Ruskey.) Can a piano player run through all possible 4-note chords that span at most one octave, changing only one finger at a time? This is the problem of generating all combinations $c_t \ldots c_1$ such that $n > c_t > \cdots > c_1 \geq 0$ and $c_t - c_1 < m$, where $t = 4$ and (a) $m = 8$, $n = 52$ if we consider only the white notes of a piano keyboard; (b) $m = 13$, $n = 88$ if we consider also the black notes.

58. [*20*] Consider the piano player's problem of exercise 57 with the additional condition that the chords don't involve adjacent notes. (In other words, $c_{j+1} > c_j + 1$ for $t > j \geq 1$. Such chords tend to be more harmonious.)

59. [*M25*] Is there a *perfect* solution to the 4-note piano player's problem, in which each step moves a finger to an *adjacent* key?

60. [*23*] Design an algorithm to generate all *bounded* compositions

$$t = r_s + \cdots + r_1 + r_0, \qquad \text{where } 0 \leq r_j \leq m_j \text{ for } s \geq j \geq 0.$$

61. [*32*] Show that all bounded compositions can be generated by changing only two of the parts at each step.

▶ **62.** [*M27*] A *contingency table* is an $m \times n$ matrix of nonnegative integers (a_{ij}) having given row sums $r_i = \sum_{j=1}^{n} a_{ij}$ and column sums $c_j = \sum_{i=1}^{m} a_{ij}$, where $r_1 + \cdots + r_m = c_1 + \cdots + c_n$.

a) Show that $2 \times n$ contingency tables are equivalent to bounded compositions.
b) What is the lexicographically largest contingency table for $(r_1, \ldots, r_m; c_1, \ldots, c_n)$, when matrix entries are read row-wise from left to right and top to bottom, namely in the order $(a_{11}, a_{12}, \ldots, a_{1n}, a_{21}, \ldots, a_{mn})$?
c) What is the lexicographically largest contingency table for $(r_1, \ldots, r_m; c_1, \ldots, c_n)$, when matrix entries are read column-wise from top to bottom and left to right, namely in the order $(a_{11}, a_{21}, \ldots, a_{m1}, a_{12}, \ldots, a_{mn})$?
d) What is the lexicographically smallest contingency table for $(r_1, \ldots, r_m; c_1, \ldots, c_n)$, in the row-wise and column-wise senses?
e) Explain how to generate all contingency tables for $(r_1, \ldots, r_m; c_1, \ldots, c_n)$ in lexicographic order.

63. [*M41*] Show that all contingency tables for $(r_1, \ldots, r_m; c_1, \ldots, c_n)$ can be generated by changing exactly four entries of the matrix at each step.

▶ **64.** [*M30*] Construct a genlex Gray cycle for all of the $2^s\binom{s+t}{t}$ *subcubes* that have s digits and t asterisks, using only the transformations $*0 \leftrightarrow 0*$, $*1 \leftrightarrow 1*$, $0 \leftrightarrow 1$. For example, one such cycle when $s = t = 2$ is

$$(00{*}{*}, 01{*}{*}, 0{*}1{*}, 0{*}{*}1, 0{*}{*}0, 0{*}0{*}, {*}00{*}, {*}01{*}, {*}0{*}1, {*}0{*}0, {*}{*}00, {*}{*}01,$$
$${*}{*}11, {*}{*}10, {*}1{*}0, {*}1{*}1, {*}11{*}, {*}10{*}, 1{*}0{*}, 1{*}{*}0, 1{*}{*}1, 1{*}1{*}, 11{*}{*}, 10{*}{*}).$$

65. [*M40*] Enumerate the total number of genlex Gray paths on subcubes that use only the transformations allowed in exercise 64. How many of those paths are cycles?

▶ **66.** [*22*] Given $n \geq t \geq 0$, show that there is a Gray path through all of the canonical bases $(\alpha_1, \ldots, \alpha_t)$ of exercise 12, changing just one bit at each step. For example, one such path when $n = 3$ and $t = 2$ is

$$\begin{matrix}001\\010\end{matrix},\ \begin{matrix}101\\010\end{matrix},\ \begin{matrix}101\\110\end{matrix},\ \begin{matrix}001\\110\end{matrix},\ \begin{matrix}001\\100\end{matrix},\ \begin{matrix}011\\100\end{matrix},\ \begin{matrix}010\\100\end{matrix}.$$

67. [*46*] Consider the Ising configurations of exercise 13 for which $a_0 = 0$. Given n, t, and r, is there a Gray cycle for these configurations in which all transitions have the forms $0^k1 \leftrightarrow 10^k$ or $01^k \leftrightarrow 1^k0$? For example, in the case $n = 9$, $t = 5$, $r = 6$, there is a unique cycle

$$(010101110,\ 010110110,\ 011010110,\ 011011010,\ 011101010,\ 010111010).$$

68. [*M01*] If α is a t-combination, what is (a) $\partial^t\alpha$? (b) $\partial^{t+1}\alpha$?

▶ **69.** [*M22*] How large is the smallest set A of t-combinations for which $|\partial A| < |A|$?

70. [*M25*] What is the maximum value of $\kappa_t N - N$, for $N \geq 0$?

71. [*M20*] How many t-cliques can a million-edge graph have?

▶ **72.** [*M22*] Show that if N has the degree-t combinatorial representation (57), there is an easy way to find the degree-s combinatorial representation of the complementary number $M = \binom{s+t}{t} - N$, whenever $N < \binom{s+t}{t}$. Derive (63) as a consequence.

73. [*M23*] (A. J. W. Hilton, 1976.) Let A be a set of s-combinations and B a set of t-combinations, both contained in $U = \{0, \ldots, n-1\}$ where $n \geq s + t$. Show that if A and B are *cross-intersecting*, in the sense that $\alpha \cap \beta \neq \emptyset$ for all $\alpha \in A$ and $\beta \in B$, then so are the sets Q_{Mns} and Q_{Nnt} defined in Theorem K, where $M = |A|$ and $N = |B|$.

74. [*M21*] What are $|\varrho P_{Nt}|$ and $|\varrho Q_{Nnt}|$ in Theorem K?

75. [*M20*] The right-hand side of (60) is not always the degree-$(t-1)$ combinatorial representation of $\kappa_t N$, because $v-1$ might be zero. Show, however, that a positive integer N has at most two representations if we allow $v=0$ in (57), and both of them yield the same value $\kappa_t N$ according to (60). Therefore

$$\kappa_k \kappa_{k+1} \ldots \kappa_t N = \binom{n_t}{k-1} + \binom{n_{t-1}}{k-2} + \cdots + \binom{n_v}{k-1+v-t} \qquad \text{for } 1 \le k \le t.$$

76. [*M20*] Find a simple formula for $\kappa_t(N+1) - \kappa_t N$.

▸ **77.** [*M26*] Prove the following properties of the κ functions by manipulating binomial coefficients, without assuming Theorem K:

a) $\kappa_t(M+N) \le \kappa_t M + \kappa_t N$.

b) $\kappa_t(M+N) \le \max(\kappa_t M, N) + \kappa_{t-1} N$.

Hint: $\binom{m_t}{t} + \cdots + \binom{m_1}{1} + \binom{n_t}{t} + \cdots + \binom{n_1}{1}$ is equal to $\binom{m_t \vee n_t}{t} + \cdots + \binom{m_1 \vee n_1}{1} + \binom{m_t \wedge n_t}{t} + \cdots + \binom{m_1 \wedge n_1}{1}$, where $\vee$ and $\wedge$ denote max and min.

78. [*M22*] Show that Theorem K follows easily from inequality (b) in the previous exercise. Conversely, both inequalities are simple consequences of Theorem K. *Hint:* Any set A of t-combinations can be written $A = A_1 + A_0 0$, where $A_1 = \{\alpha \in A \mid 0 \notin \alpha\}$.

79. [*M23*] Prove that if $t \ge 2$, we have $M \ge \mu_t N$ if and only if $M + \lambda_{t-1} M \ge N$.

80. [*HM26*] (L. Lovász, 1979.) The function $\binom{x}{t}$ increases monotonically from 0 to ∞ as x increases from $t-1$ to ∞; hence we can define

$$\underline{\kappa}_t N = \binom{x}{t-1}, \qquad \text{if } N = \binom{x}{t} \text{ and } x \ge t-1.$$

Prove that $\kappa_t N \ge \underline{\kappa}_t N$ for all integers $t \ge 1$ and $N \ge 0$. *Hint:* Equality holds when x is an integer.

▸ **81.** [*M27*] Show that the minimum shadow sizes in Theorem M are given by (64).

82. [*HM31*] The Takagi function of Fig. 27 is defined for $0 \le x \le 1$ by the formula

$$\tau(x) = \sum_{k=1}^{\infty} \int_0^x r_k(t)\, dt,$$

where $r_k(t) = (-1)^{\lfloor 2^k t \rfloor}$ is the Rademacher function of Eq. 7.2.1.1–(16).

a) Prove that $\tau(x)$ is continuous in the interval $[0\,..\,1]$, but its derivative does not exist at any point.

b) Show that $\tau(x)$ is the only continuous function that satisfies

$$\tau(\tfrac{1}{2}x) \;=\; \tau(1-\tfrac{1}{2}x) \;=\; \tfrac{1}{2}x + \tfrac{1}{2}\tau(x) \qquad \text{for } 0 \le x \le 1.$$

c) What is the asymptotic value of $\tau(\epsilon)$ when ϵ is small?

d) Prove that $\tau(x)$ is rational when x is rational.

e) Find all roots of the equation $\tau(x) = 1/2$.

f) Find all roots of the equation $\tau(x) = \max_{0 \le x \le 1} \tau(x)$.

83. [*HM46*] Determine the set R of all rational numbers r such that the equation $\tau(x) = r$ has uncountably many solutions. If $\tau(x)$ is rational and x is irrational, is it true that $\tau(x) \in R$? (*Warning:* This problem can be addictive.)

84. [*HM27*] If $T = \binom{2t-1}{t}$, prove the asymptotic formula

$$\kappa_t N - N \;=\; \frac{T}{t}\Bigl(\tau\Bigl(\frac{N}{T}\Bigr) + O\Bigl(\frac{(\log t)^3}{t}\Bigr)\Bigr) \qquad \text{for } 0 \le N \le T.$$

85. [*HM21*] Relate the functions $\lambda_t N$ and $\mu_t N$ to the Takagi function $\tau(x)$.

86. [*M20*] Prove the law of spread/core duality, $X^{\sim +} = X^{\circ\sim}$.

87. [*M21*] True or false: (a) $X \subseteq Y^\circ$ if and only if $Y^\sim \subseteq X^{\sim\circ}$; (b) $X^{\circ+\circ} = X^\circ$; (c) $\alpha M \le N$ if and only if $M \le \beta N$.

88. [*M20*] Explain why cross order is useful, by completing the proof of Lemma S.

89. [*16*] Compute the α and β functions for the $2 \times 2 \times 3$ torus (69).

90. [*M22*] Prove the basic compression lemma, (85).

91. [*M24*] Prove Theorem W for two-dimensional toruses $T(l, m)$, $l \le m$.

92. [*M28*] Let $x = x_1 \ldots x_{n-1}$ be the Nth element of the torus $T(m_1, \ldots, m_{n-1})$, and let S be the set of all elements of $T(m_1, \ldots, m_{n-1}, m)$ that are $\preceq x_1 \ldots x_{n-1}(m{-}1)$ in cross order. If N_a elements of S have final component a, for $0 \le a < m$, prove that $N_{m-1} = N$ and $N_{a-1} = \alpha N_a$ for $1 \le a < m$, where α is the spread function for standard sets in $T(m_1, \ldots, m_{n-1})$.

93. [*M25*] (a) Find an N for which the conclusion of Theorem W is false when the parameters $m_1, m_2, \ldots, m_n$ have not been sorted into nondecreasing order. (b) Where does the proof of that theorem use the hypothesis that $m_1 \le m_2 \le \cdots \le m_n$?

94. [*M20*] Show that the ∂ half of Corollary C follows from the ϱ half. *Hint:* The complements of the multicombinations (92) with respect to U are 3211, 3210, 3200, 3110, 3100, 3000, 2110, 2100, 2000, 1100, 1000.

95. [*17*] Explain why Theorems K and M follow from Corollary C.

▶ **96.** [*M22*] If S is an infinite sequence $(s_0, s_1, s_2, \ldots)$ of positive integers, let

$$\binom{S(n)}{k} = [z^k] \prod_{j=0}^{n-1} (1 + z + \cdots + z^{s_j});$$

thus $\binom{S(n)}{k}$ is the ordinary binomial coefficient $\binom{n}{k}$ if $s_0 = s_1 = s_2 = \cdots = 1$.

Generalizing the combinatorial number system, show that every nonnegative integer N has a unique representation

$$N = \binom{S(n_t)}{t} + \binom{S(n_{t-1})}{t-1} + \cdots + \binom{S(n_1)}{1}$$

where $n_t \ge n_{t-1} \ge \cdots \ge n_1 \ge 0$ and $\{n_t, n_{t-1}, \ldots, n_1\} \subseteq \{s_0 \cdot 0, s_1 \cdot 1, s_2 \cdot 2, \ldots\}$. Use this representation to give a simple formula for the numbers $|\partial P_{Nt}|$ in Corollary C.

▶ **97.** [*M26*] The text remarked that the vertices of a convex polyhedron can be perturbed slightly so that all of its faces are simplexes. In general, any set of combinations that contains the shadows of all its elements is called a *simplicial complex*; thus C is a simplicial complex if and only if $\alpha \subseteq \beta$ and $\beta \in C$ implies that $\alpha \in C$, if and only if C is an order ideal with respect to set inclusion.

The *size vector* of a simplicial complex C on n vertices is $(N_0, N_1, \ldots, N_n)$ when C contains exactly N_t combinations of size t.

a) What are the size vectors of the five regular solids (the tetrahedron, cube, octahedron, dodecahedron, and icosahedron), when their vertices are slightly tweaked?

b) Construct a simplicial complex with size vector $(1, 4, 5, 2, 0)$.
c) Find a necessary and sufficient condition that a given size vector $(N_0, N_1, \ldots, N_n)$ is feasible.
d) Prove that $(N_0, \ldots, N_n)$ is feasible if and only its "dual" vector $(\overline{N}_0, \ldots, \overline{N}_n)$ is feasible, where we define $\overline{N}_t = \binom{n}{t} - N_{n-t}$.
e) List all feasible size vectors $(N_0, N_1, N_2, N_3, N_4)$ and their duals. Which of them are self-dual?

98. [*30*] Continuing exercise 97, find an efficient way to count the feasible size vectors $(N_0, N_1, \ldots, N_n)$ when $n \le 100$.

99. [*M25*] A *clutter* is a set C of combinations that are incomparable, in the sense that $\alpha \subseteq \beta$ and $\alpha, \beta \in C$ implies $\alpha = \beta$. The size vector of a clutter is defined as in exercise 97.
a) Find a necessary and sufficient condition that $(M_0, M_1, \ldots, M_n)$ is the size vector of a clutter.
b) List all such size vectors in the case $n = 4$.

▸ **100.** [*M30*] (Clements and Lindström.) Let A be a "simplicial multicomplex," a set of submultisets of the multiset U in Corollary C with the property that $\partial A \subseteq A$. How large can the total weight $\nu A = \sum\{|\alpha| \mid \alpha \in A\}$ be when $|A| = N$?

101. [*M25*] If $f(x_1, \ldots, x_n)$ is a Boolean formula, let $F(p)$ be the probability that $f(x_1, \ldots, x_n) = 1$ when each variable x_j independently is 1 with probability p.
a) Calculate $G(p)$ and $H(p)$ for the Boolean formulas $g(w, x, y, z) = wxz \vee wyz \vee xy\bar{z}$, $h(w, x, y, z) = \bar{w}yz \vee xyz$.
b) Show that there is a *monotone* Boolean function $f(w, x, y, z)$ such that $F(p) = G(p)$, but there is no such function with $F(p) = H(p)$. Explain how to test this condition in general.

102. [*HM35*] (F. S. Macaulay, 1927.) A *polynomial ideal* I in the variables $\{x_1 \ldots, x_s\}$ is a set of polynomials closed under the operations of addition, multiplication by a constant, and multiplication by any of the variables. It is called *homogeneous* if it consists of all linear combinations of a set of homogeneous polynomials, namely of polynomials like $xy + z^2$ whose terms all have the same degree. Let N_t be the maximum number of linearly independent elements of degree t in I. For example, if $s = 2$, the set of all $\alpha(x_0, x_1, x_2)(x_0 x_1^2 - 2x_1 x_2^2) + \beta(x_0, x_1, x_2) x_0 x_1 x_2^2$, where α and β run through all possible polynomials in $\{x_0, x_1, x_2\}$, is a homogeneous polynomial ideal with $N_0 = N_1 = N_2 = 0$, $N_3 = 1$, $N_4 = 4$, $N_5 = 9$, $N_6 = 15$, ...
a) Prove that for any such ideal I there is another ideal I' in which all homogeneous polynomials of degree t are linear combinations of N_t independent *monomials*. (A monomial is a product of variables, like $x_1^3 x_2 x_5^4$.)
b) Use Theorem M and (64) to prove that $N_{t+1} \ge N_t + \kappa_s N_t$ for all $t \ge 0$.
c) Show that $N_{t+1} > N_t + \kappa_s N_t$ occurs for only finitely many t. (This statement is equivalent to "Hilbert's basis theorem," proved by David Hilbert in *Göttinger Nachrichten* (1888), 450–457; *Math. Annalen* **36** (1890), 473–534.)

▸ **103.** [*M38*] The shadow of a subcube $a_1 \ldots a_n$, where each a_j is either 0 or 1 or $*$, is obtained by replacing some $*$ by 0 or 1. For example,

$$\partial 0{*}11{*}0 \;=\; \{0011{*}0,\ 0111{*}0,\ 0{*}1100,\ 0{*}1110\}.$$

Find a set P_{Nst} such that, if A is any set of N subcubes $a_1 \ldots a_n$ having s digits and t asterisks, $|\partial A| \ge |P_{Nst}|$.

104. [*M41*] The shadow of a binary string $a_1 \ldots a_n$ is obtained by deleting one of its bits. For example,

$$\partial 110010010 \;=\; \{10010010,\, 11010010,\, 11000010,\, 11001000,\, 11001001\}.$$

Find a set P_{Nn} such that, if A is any set of N binary strings $a_1 \ldots a_n$, $|\partial A| \geq |P_{Nn}|$.

105. [*M20*] A *universal cycle of t-combinations* for $\{0, 1, \ldots, n-1\}$ is a cycle of $\binom{n}{t}$ numbers whose blocks of t consecutive elements run through every t-combination $\{c_1, \ldots, c_t\}$. For example,

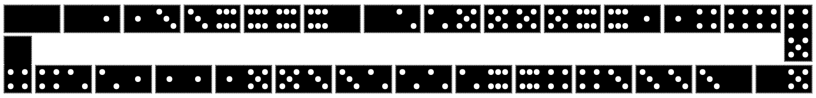
$$(02145061320516243152630425364103546)$$

is a universal cycle when $t = 3$ and $n = 7$.

Prove that no such cycle is possible unless $\binom{n}{t}$ is a multiple of n.

106. [*M21*] (L. Poinsot, 1809.) Find a "nice" universal cycle of 2-combinations for $\{0, 1, \ldots, 2m\}$. *Hint:* Consider the differences of consecutive elements, mod $(2m+1)$.

107. [*22*] (O. Terquem, 1849.) Poinsot's theorem implies that all 28 dominoes of a traditional "double-six" set can be arranged in a cycle so that the spots of adjacent dominoes match each other:



How many such cycles are possible?

108. [*M31*] Find universal cycles of 3-combinations for the sets $\{0, \ldots, n-1\}$ when $n \bmod 3 \neq 0$.

109. [*M31*] Find universal cycles of 3-*multicombinations* for $\{0, 1, \ldots, n-1\}$ when $n \bmod 3 \neq 0$ (namely for combinations $d_1 d_2 d_3$ with repetitions permitted). For example,

$$(00012241112330222344133340024440113)$$

is such a cycle when $n = 5$.

▶ **110.** [*26*] *Cribbage* is a game played with 52 cards, where each card has a suit ($\clubsuit$, $\diamondsuit$, $\heartsuit$, or $\spadesuit$) and a face value (`A`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`, `10`, `J`, `Q`, or `K`). One feature of the game is to compute the score of a 5-card combination $C = \{c_1, c_2, c_3, c_4, c_5\}$, where one card c_k is called the *starter*. The score is the sum of points computed as follows, for each subset S of C and each choice of k: Let $|S| = s$.

i) Fifteens: If $\sum\{v(c) \mid c \in S\} = 15$, where $\bigl(v(\mathtt{A}), v(\mathtt{2}), v(\mathtt{3}), \ldots, v(\mathtt{9}), v(\mathtt{10}), v(\mathtt{J}), v(\mathtt{Q}), v(\mathtt{K})\bigr) = (1, 2, 3, \ldots, 9, 10, 10, 10, 10)$, score two points.

ii) Pairs: If $s = 2$ and both cards have the same face value, score two points.

iii) Runs: If $s \geq 3$ and the face values are consecutive, and if C does not contain a run of length $s+1$, score s points.

iv) Flushes: If $s = 4$ and all cards of S have the same suit, and if $c_k \notin S$, score $4 + [c_k$ has the same suit as the others$]$.

v) Nobs: If $s = 1$ and $c_k \notin S$, score 1 if the card is `J` of the same suit as c_k.

For example, if you hold $\{\mathtt{J}\clubsuit, \mathtt{5}\clubsuit, \mathtt{5}\diamondsuit, \mathtt{6}\heartsuit\}$ and if $\mathtt{4}\clubsuit$ is the starter, you score 4×2 for fifteens, 2 for a pair, 2×3 for runs, plus 1 for nobs, totalling 17.

Exactly how many combinations and starter choices lead to a score of x points, for $x = 0, 1, 2, \ldots$?

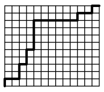
SECTION 7.2.1.3

1. Given a multiset, form the sequence $e_t \ldots e_2 e_1$ from right to left by listing the distinct elements first, then those that appear twice, then those that appear thrice, etc. Let us set $e_{-j} \leftarrow s-j$ for $0 \le j \le s = n-t$, so that every element e_j for $1 \le j \le t$ is equal to some element to its right in the sequence $e_t \ldots e_1 e_0 \ldots e_{-s}$. If the first such element is $e_{c_j - s}$, we obtain a solution to (3). Conversely, every solution to (3) yields a unique multiset $\{e_1, \ldots, e_t\}$, because $c_j < s+j$ for $1 \le j \le t$.

[A similar correspondence was proposed by E. Catalan: If $0 \le e_1 \le \cdots \le e_t \le s$, let

$$\{c_1, \ldots, c_t\} \;=\; \{e_1, \ldots, e_t\} \cup \{s+j \mid 1 \le j < t \text{ and } e_j = e_{j+1}\}.$$

See *Mémoires de la Soc. roy. des Sciences de Liège* (2) **12** (1885), *Mélanges Math.*, 3.]

2. Start at the bottom left corner; then go up for each 0, go right for each 1. The result is .

3. In this algorithm, variable r is the least positive index such that $q_r > 0$.

F1. [Initialize.] Set $q_j \leftarrow 0$ for $1 \le j \le t$, and $q_0 \leftarrow s$. (We assume that $st > 0$.)

F2. [Visit.] Visit the composition $q_t \ldots q_0$. Go to F4 if $q_0 = 0$.

F3. [Easy case.] Set $q_0 \leftarrow q_0 - 1$, $r \leftarrow 1$, and go to F5.

F4. [Tricky case.] Terminate if $r = t$. Otherwise set $q_0 \leftarrow q_r - 1$, $q_r \leftarrow 0$, $r \leftarrow r+1$.

F5. [Increase q_r.] Set $q_r \leftarrow q_r + 1$ and return to F2. ▮

[See *CACM* **11** (1968), 430; **12** (1969), 187. The task of generating such compositions in *decreasing* lexicographic order is more difficult.]

4. We can reverse the roles of 0 and 1 in (14), so that $0^{q_t} 1 0^{q_{t-1}} 1 \ldots 1 0^{q_1} 1 0^{q_0} = 1^{r_s} 0 1^{r_{s-1}} 0 \ldots 0 1^{r_1} 0 1^{r_0}$. This gives $0^1 1 0^0 1 0^2 1 0^2 1 0^4 1 0^0 1 0^0 1 0^0 1 0^0 1 0^0 1 0^1 1 0^0 1 0^1 1 0^0 = 1^0 0 1^2 0 1^0 0 1^1 0 1^0 0 1^1 0 1^0 0 1^0 0 1^0 0 1^6 0 1^2 0 1^1$. Lexicographic order of $a_{n-1} \ldots a_1 a_0$ corresponds to lexicographic order of $r_s \ldots r_1 r_0$.

Incidentally, there's also a multiset connection: $\{d_t, \ldots, d_1\} = \{r_s \cdot s, \ldots, r_0 \cdot 0\}$. For example, $\{10, 10, 8, 6, 2, 2, 2, 2, 2, 2, 1, 1, 0\} = \{0 \cdot 11, 2 \cdot 10, 0 \cdot 9, 1 \cdot 8, 0 \cdot 7, 1 \cdot 6, 0 \cdot 5, 0 \cdot 4, 0 \cdot 3, 6 \cdot 2, 2 \cdot 1, 1 \cdot 0\}$.

5. (a) Set $x_j = c_j - \lfloor (j-1)/2 \rfloor$ in each t-combination of $n + \lfloor t/2 \rfloor$. (b) Set $x_j = c_j + j + 1$ in each t-combination of $n - t - 2$.

(A similar approach finds all solutions $(x_t, \ldots, x_1)$ to the inequalities $x_{j+1} \ge x_j + \delta_j$ for $0 \le j \le t$, given the values of x_{t+1}, $(\delta_t, \ldots, \delta_1)$, and x_0.)

6. Assume that $t > 0$. We get to T3 when $c_1 > 0$; to T5 when $c_2 = c_1 + 1 > 1$; to T4 for $2 \le j \le t+1$ when $c_j = c_1 + j - 1 \ge j$. So the counts are: T1, 1; T2, $\binom{n}{t}$; T3, $\binom{n-1}{t}$; T4, $\binom{n-2}{t-1} + \binom{n-2}{t-2} + \cdots + \binom{n-t-1}{0} = \binom{n-1}{t-1}$; T5, $\binom{n-2}{t-1}$; T6, $\binom{n-1}{t-1} + \binom{n-2}{t-1} - 1$.

7. A procedure slightly simpler than Algorithm T suffices: Assume that $s < n$.

S1. [Initialize.] Set $b_j \leftarrow j + n - s - 1$ for $1 \le j \le s$; then set $j \leftarrow 1$.

S2. [Visit.] Visit the combination $b_s \ldots b_2 b_1$. Terminate if $j > s$.

S3. [Decrease b_j.] Set $b_j \leftarrow b_j - 1$. If $b_j < j$, set $j \leftarrow j + 1$ and return to S2.

S4. [Reset $b_{j-1} \ldots b_1$.] While $j > 1$, set $b_{j-1} \leftarrow b_j - 1$, $j \leftarrow j - 1$, and repeat until $j = 1$. Go to S2. ▮

(See S. Dvořák, *Comp. J.* **33** (1990), 188. Notice that if $x_k = n - b_k$ for $1 \le k \le s$, this algorithm runs through all combinations $x_s \dots x_2x_1$ of $\{1,2,\dots,n\}$ with $1 \le x_s < \cdots < x_2 < x_1 \le n$, in *increasing* lexicographic order.)

8. A1. [Initialize.] Set $a_n \dots a_0 \leftarrow 0^{s+1}1^t$, $q \leftarrow t$, $r \leftarrow 0$. (We assume that $0 < t < n$.)

A2. [Visit.] Visit the combination $a_{n-1} \dots a_1a_0$. Go to A4 if $q = 0$.

A3. [Replace $\dots 01^q$ by $\dots 101^{q-1}$.] Set $a_q \leftarrow 1$, $a_{q-1} \leftarrow 0$, $q \leftarrow q-1$; then if $q = 0$, set $r \leftarrow 1$. Return to A2.

A4. [Shift block of 1s.] Set $a_r \leftarrow 0$ and $r \leftarrow r+1$. Then if $a_r = 1$, set $a_q \leftarrow 1$, $q \leftarrow q+1$, and repeat step A4.

A5. [Carry to left.] Terminate if $r = n$; otherwise set $a_r \leftarrow 1$.

A6. [Odd?] If $q > 0$, set $r \leftarrow 0$. Return to A2. ▌

In step A2, q and r point respectively to the rightmost 0 and 1 in $a_{n-1} \dots a_0$. Steps A1, …, A6 are executed with frequency 1, $\binom{n}{t}$, $\binom{n-1}{t-1}$, $\binom{n}{t} - 1$, $\binom{n-1}{t}$, $\binom{n-1}{t} - 1$.

9. (a) The first $\binom{n-1}{t}$ strings begin with 0 and have $2A_{(s-1)t}$ bit changes; the other $\binom{n-1}{t-1}$ begin with 1 and have $2A_{s(t-1)}$. And $\nu(01^t0^{s-1} \oplus 10^s1^{t-1}) = 2\min(s,t)$.

(b) Solution 1 (direct): Let $B_{st} = A_{st} + \min(s,t) + 1$. Then

$$B_{st} = B_{(s-1)t} + B_{s(t-1)} + [s=t] \quad \text{when } st > 0; \qquad B_{st} = 1 \quad \text{when } st = 0.$$

Consequently $B_{st} = \sum_{k=0}^{\min(s,t)} \binom{s+t-2k}{s-k}$. If $s \le t$ this is $\le \sum_{k=0}^{s} \binom{s+t-k}{s-k} = \binom{s+t+1}{s} = \binom{s+t}{s}\frac{s+t+1}{t+1} < 2\binom{s+t}{t}$.

Solution 2 (indirect): The algorithm in answer 8 makes $2(x+y)$ bit changes when steps (A3, A4) are executed (x,y) times. Thus $A_{st} \le \binom{n-1}{t-1} + \binom{n}{t} - 1 < 2\binom{n}{t}$.

[The comment in answer 7.2.1.1–3 therefore applies to combinations as well.]

10. Each scenario corresponds to a $(4,4)$-combination $b_4b_3b_2b_1$ or $c_4c_3c_2c_1$ in which A wins games $\{8-b_4, 8-b_3, 8-b_2, 8-b_1\}$ and N wins games $\{8-c_4, 8-c_3, 8-c_2, 8-c_1\}$, because we can assume that the losing team wins the remaining games in a series of 8. (Equivalently, we can generate all permutations of $\{\mathrm{A,A,A,A,N,N,N,N}\}$ and omit the trailing run of As or Ns.) The American League wins if and only if $b_1 \ne 0$, if and only if $c_1 = 0$. The formula $\binom{c_4}{4} + \binom{c_3}{3} + \binom{c_2}{2} + \binom{c_1}{1}$ assigns a unique integer between 0 and 69 to each scenario.

For example, ANANAA $\iff a_7 \dots a_1a_0 = 01010011 \iff b_4b_3b_2b_1 = 7532 \iff c_4c_3c_2c_1 = 6410$, and this is the scenario of rank $\binom{6}{4} + \binom{4}{3} + \binom{1}{2} + \binom{0}{1} = 19$ in lexicographic order. (Notice that the term $\binom{c_j}{j}$ will be zero if and only if it corresponds to a trailing N.)

11. AAAA (9 times), NNNN (8), and ANAAA (7) were most common. Exactly 27 of the 70 failed to occur, including all four beginning with NNNA. (We disregard the games that were tied because of darkness, in 1907, 1912, and 1922. The case ANNAAA should perhaps be excluded too, because it occurred only in 1920 as part of ANNAAAA in a best-of-nine series. The scenario NNAAANN occurred for the first time in 2001.)

12. (a) Let V_j be the subspace $\{a_{n-1} \dots a_0 \in V \mid a_k = 0 \text{ for } 0 \le k < j\}$, so that $\{0\dots0\} = V_n \subseteq V_{n-1} \subseteq \cdots \subseteq V_0 = V$. Then $\{c_1, \dots, c_t\} = \{c \mid V_c \ne V_{c+1}\}$, and α_k is the unique element $a_{n-1} \dots a_0$ of V with $a_{c_j} = [j=k]$ for $1 \le j \le t$.

Incidentally, the $t \times n$ matrix corresponding to a canonical basis is said to be in *reduced row-echelon form*. It can be found by a standard "triangulation" algorithm (see exercise 4.6.1–19 and Algorithm 4.6.2N).

(b) The 2-nomial coefficient $\binom{n}{t}_2 = 2^t\binom{n-1}{t}_2 + \binom{n-1}{t-1}_2$ of exercise 1.2.6–58 has the right properties, because $2^t\binom{n-1}{t}_2$ binary vector spaces have $c_t < n-1$ and $\binom{n-1}{t-1}_2$ have $c_t = n-1$. [In general the number of canonical bases with r asterisks is the number of partitions of r into at most t parts, with no part exceeding $n-t$, and this is $[z^r]\binom{n}{t}_z$ by Eq. 7.2.1.4–(51). See D. E. Knuth, *J. Combinatorial Theory* **10** (1971), 178–180.]

(c) The following algorithm assumes that $n > t > 0$ and that $a_{(t+1)j} = 0$ for $t \le j \le n$.

V1. [Initialize.] Set $a_{kj} \leftarrow [j = k-1]$ for $1 \le k \le t$ and $0 \le j < n$. Also set $q \leftarrow t$, $r \leftarrow 0$.

V2. [Visit.] (At this point we have $a_{k(k-1)} = 1$ for $1 \le k \le q$, $a_{(q+1)q} = 0$, and $a_{1r} = 1$.) Visit the canonical basis $(a_{1(n-1)} \ldots a_{11}a_{10}, \ldots, a_{t(n-1)} \ldots a_{t1}a_{t0})$. Go to V4 if $q > 0$.

V3. [Find block of 1s.] Set $q \leftarrow 1, 2, \ldots,$ until $a_{(q+1)(q+r)} = 0$. Terminate if $q + r = n$.

V4. [Add 1 to column $q+r$.] Set $k \leftarrow 1$. If $a_{k(q+r)} = 1$, set $a_{k(q+r)} \leftarrow 0$, $k \leftarrow k+1$, and repeat until $a_{k(q+r)} = 0$. Then if $k \le q$, set $a_{k(q+r)} \leftarrow 1$; otherwise set $a_{q(q+r)} \leftarrow 1$, $a_{q(q+r-1)} \leftarrow 0$, $q \leftarrow q-1$.

V5. [Shift block right.] If $q = 0$, set $r \leftarrow r+1$. Otherwise, if $r > 0$, set $a_{k(k-1)} \leftarrow 1$ and $a_{k(r+k-1)} \leftarrow 0$ for $1 \le k \le q$, then set $r \leftarrow 0$. Go to V2. ▌

Step V2 finds $q > 0$ with probability $1 - (2^{n-t}-1)/(2^n-1) \approx 1 - 2^{-t}$, so we could save time by treating this case separately.

(d) Since $999999 = 4\binom{8}{4}_2 + 16\binom{7}{4}_2 + 5\binom{6}{3}_2 + 5\binom{5}{3}_2 + 8\binom{4}{3}_2 + 0\binom{3}{2}_2 + 4\binom{2}{2}_2 + 1\binom{1}{1}_2 + 2\binom{0}{1}_2$, the millionth output has binary columns 4, 16/2, 5, 5, 8/2, 0, 4/2, 1, 2/2, namely

$$\begin{aligned}
\alpha_1 &= 0\,0\,1\,1\,0\,0\,0\,1\,1,\\
\alpha_2 &= 0\,0\,0\,0\,0\,0\,1\,0\,0,\\
\alpha_3 &= 1\,0\,1\,1\,1\,0\,0\,0\,0,\\
\alpha_4 &= 0\,1\,0\,0\,0\,0\,0\,0\,0.
\end{aligned}$$

[*Reference:* E. Calabi and H. S. Wilf, *J. Combinatorial Theory* **A22** (1977), 107–109.]

13. Let $n = s + t$. There are $\binom{s-1}{\lceil (r-1)/2\rceil}\binom{t-1}{\lfloor (r-1)/2\rfloor}$ configurations beginning with 0 and $\binom{s-1}{\lfloor (r-1)/2\rfloor}\binom{t-1}{\lceil (r-1)/2\rceil}$ beginning with 1, because an Ising configuration that begins with 0 corresponds to a composition of s 0s into $\lceil (r+1)/2\rceil$ parts and a composition of t 1s into $\lfloor (r+1)/2\rfloor$ parts. We can generate all such pairs of compositions and weave them into configurations. [See E. Ising, *Zeitschrift für Physik* **31** (1925), 253–258; J. M. S. Simões Pereira, *CACM* **12** (1969), 562.]

14. Start with $l[j] \leftarrow j-1$ and $r[j-1] \leftarrow j$ for $1 \le j \le n$; $l[0] \leftarrow n$, $r[n] \leftarrow 0$. To get the next combination, assuming that $t > 0$, set $p \leftarrow s$ if $l[0] > s$, otherwise $p \leftarrow r[n]-1$. Terminate if $p \le 0$; otherwise set $q \leftarrow r[p]$, $l[q] \leftarrow l[p]$, and $r[l[p]] \leftarrow q$. Then if $r[q] > s$ and $p < s$, set $r[p] \leftarrow r[n]$, $l[r[n]] \leftarrow p$, $r[s] \leftarrow r[q]$, $l[r[q]] \leftarrow s$, $r[n] \leftarrow 0$, $l[0] \leftarrow n$; otherwise set $r[p] \leftarrow r[q]$, $l[r[q]] \leftarrow p$. Finally set $r[q] \leftarrow p$ and $l[p] \leftarrow q$.

[See Korsh and Lipschutz, *J. Algorithms* **25** (1997), 321–335, where the idea is extended to a loopless algorithm for multiset permutations. *Caution:* This exercise, like exercise 7.2.1.1–16, is more academic than practical, because the routine that visits the linked list might need a loop that nullifies any advantage of loopless generation.]

15. (The stated fact is true because lexicographic order of $c_t \ldots c_1$ corresponds to lexicographic order of $a_{n-1} \ldots a_0$, which is reverse lexicographic order of the complementary sequence $1 \ldots 1 \oplus a_{n-1} \ldots a_0$.) By Theorem L, the combination $c_t \ldots c_1$ is visited *before* exactly $\binom{b_s}{s} + \cdots + \binom{b_2}{2} + \binom{b_1}{1}$ others have been visited, and we must have

$$\binom{b_s}{s} + \cdots + \binom{b_1}{1} + \binom{c_t}{t} + \cdots + \binom{c_1}{1} \;=\; \binom{s+t}{t} - 1.$$

This general identity can be written

$$\sum_{j=0}^{n-1} x_j \binom{j}{x_0 + \cdots + x_j} + \sum_{j=0}^{n-1} \bar{x}_j \binom{j}{\bar{x}_0 + \cdots + \bar{x}_j} \;=\; \binom{n}{x_0 + \cdots + x_{n-1}} - 1$$

when each x_j is 0 or 1, and $\bar{x}_j = 1 - x_j$; it follows also from the equation

$$x_n \binom{n}{x_0 + \cdots + x_n} + \bar{x}_n \binom{n}{\bar{x}_0 + \cdots + \bar{x}_n} \;=\; \binom{n+1}{x_0 + \cdots + x_n} - \binom{n}{x_0 + \cdots + x_{n-1}}.$$

16. Since $999999 = \binom{1414}{2} + \binom{1008}{1} = \binom{182}{3} + \binom{153}{2} + \binom{111}{1} = \binom{71}{4} + \binom{56}{3} + \binom{36}{2} + \binom{14}{1} = \binom{43}{5} + \binom{32}{4} + \binom{21}{3} + \binom{15}{2} + \binom{6}{1}$, the answers are (a) 1414 1008; (b) 182 153 111; (c) 71 56 36 14; (d) 43 32 21 15 6; (e) 1000000 999999 ... 2 0.

17. By Theorem L, n_t is the largest integer such that $N \ge \binom{n_t}{t}$; the remaining terms are the degree-$(t-1)$ representation of $N - \binom{n_t}{t}$.

A simple sequential method for $t > 1$ starts with $x = 1$, $c = t$, and sets $c \leftarrow c + 1$, $x \leftarrow xc/(c-t)$ zero or more times until $x > N$; then we complete the first phase by setting $x \leftarrow x(c-t)/c$, $c \leftarrow c - 1$, at which point we have $x = \binom{c}{t} \le N < \binom{c+1}{t}$. Set $n_t \leftarrow c$, $N \leftarrow N - x$; terminate with $n_1 \leftarrow N$ if $t = 2$; otherwise set $x \leftarrow xt/c$, $t \leftarrow t-1$, $c \leftarrow c - 1$; while $x > N$ set $x \leftarrow x(c-t)/c$, $c \leftarrow c - 1$; repeat. This method requires $O(n)$ arithmetic operations if $N < \binom{n}{t}$, so it is suitable unless t is small and N is large.

When $t = 2$, exercise 1.2.4–41 tells us that $n_2 = \lfloor \sqrt{2N+2} + \frac{1}{2} \rfloor$. In general, n_t is $\lfloor x \rfloor$ where x is the largest root of $x^{\underline{t}} = t!\,N$; this root can be approximated by reverting the series $y = (x^{\underline{t}})^{1/t} = x - \frac{1}{2}(t-1) + \frac{1}{24}(t^2-1)x^{-1} + \cdots$ to get $x = y + \frac{1}{2}(t-1) + \frac{1}{24}(t^2-1)/y + O(y^{-3})$. Setting $y = (t!\,N)^{1/t}$ in this formula gives a good approximation, after which we can check that $\binom{\lfloor x \rfloor}{t} \le N < \binom{\lfloor x \rfloor + 1}{t}$ or make a final adjustment. [See A. S. Fraenkel and M. Mor, *Comp. J.* **26** (1983), 336–343.]

18. A complete binary tree of $2^n - 1$ nodes is obtained, with an extra node at the top, like the "tree of losers" in replacement selection sorting (Fig. 63 in Section 5.4.1). Therefore explicit links aren't necessary; the right child of node k is node $2k+1$, and the left sibling is node $2k$, for $1 \le k < 2^{n-1}$.

This representation of a binomial tree has the curious property that node $k = (0^a 1 \alpha)_2$ corresponds to the combination whose binary string is $0^a 1 \alpha^R$.

19. It is post(1000000), where $\mathrm{post}(n) = 2^k + \mathrm{post}(n - 2^k + 1)$ if $2^k \le n < 2^{k+1}$, and $\mathrm{post}(0) = 0$. So it is 11110100001001000100.

20. $f(z) = (1 + z^{w_{n-1}}) \ldots (1 + z^{w_1})/(1-z)$, $g(z) = (1 + z^{w_0})f(z)$, $h(z) = z^{w_0} f(z)$.

21. The rank of $c_t \ldots c_2 c_1$ is $\binom{c_t+1}{t} - 1$ minus the rank of $c_{t-1} \ldots c_2 c_1$. [See H. Lüneburg, *Abh. Math. Sem. Hamburg* **52** (1982), 208–227.]

22. Since $999999 = \binom{1415}{2} - \binom{406}{1} = \binom{183}{3} - \binom{98}{2} + \binom{21}{1} = \binom{72}{4} - \binom{57}{3} + \binom{32}{2} - \binom{27}{1} = \binom{44}{5} - \binom{40}{4} + \binom{33}{3} - \binom{13}{2} + \binom{3}{1}$, the answers are (a) 1414 405; (b) 182 97 21; (c) 71 56 31 26; (d) 43 39 32 12 3; (e) 1000000 999999 999998 999996 ... 0.

23. There are $\binom{n-r}{t-r}$ combinations with $j > r$, for $r = 1, 2, \ldots, t$. (If $r = 1$ we have $c_2 = c_1 + 1$; if $r = 2$ we have $c_1 = 0$, $c_2 = 1$; if $r = 3$ we have $c_1 = 0$, $c_2 = 1$, $c_4 = c_3 + 1$; etc.) Thus the mean is $\bigl(\binom{n}{t} + \binom{n-1}{t-1} + \cdots + \binom{n-t}{0}\bigr)/\binom{n}{t} = \binom{n+1}{t}/\binom{n}{t} = (n+1)/(n+1-t)$. The average running time per step is approximately proportional to this quantity; thus the algorithm is quite fast when t is small, but slow if t is near n.

24. In fact $j_k - 2 \le j_{k+1} \le j_k + 1$ when $j_k \equiv t$ (modulo 2) and $j_k - 1 \le j_{k+1} \le j_k + 2$ when $j_k \not\equiv t$, because R5 is performed only when $c_i = i - 1$ for $1 \le i < j$.

Thus we could say, "If $j \ge 4$, set $j \leftarrow j - 1 - [j \text{ odd}]$ and go to R5" at the end of R2, if t is odd; "If $j \ge 3$, set $j \leftarrow j - 1 - [j \text{ even}]$ and go to R5" if t is even. The algorithm will then be loopless, since R4 and R5 will be performed at most twice per visit.

25. Assume that $N > N'$ and $N - N'$ is minimum; furthermore let t and c_t be minimum, subject to those assumptions. Then $c_t > c'_t$.

If there is an element $x \notin C \cup C'$ with $0 \le x < c_t$, map each t-combination of $C \cup C'$ by changing $j \mapsto j - 1$ for $j > x$; or, if there is an element $x \in C \cap C'$, map each t-combination that contains x into a $(t-1)$-combination by omitting x and changing $j \mapsto x - j$ for $j < x$. In either case the mapping preserves alternating lexicographic order; hence $N - N'$ must exceed the number of combinations between the images of C and C'. But c_t is minimum, so no such x can exist. Consequently $t = m$ and $c_t = 2m - 1$.

Now if $c'_m < c_m - 1$, we could decrease $N - N'$ by increasing c'_m. Therefore $c'_m = 2m-2$, and the problem has been reduced to finding the *maximum* of $\operatorname{rank}(c_{m-1} \ldots c_1) - \operatorname{rank}(c'_{m-1} \ldots c'_1)$, where rank is calculated as in (30).

Let $f(s,t) = \max(\operatorname{rank}(b_s \ldots b_1) - \operatorname{rank}(c_t \ldots c_1))$ over all $\{b_s, \ldots, b_1, c_t \ldots, c_1\} = \{0, \ldots, s+t-1\}$. Then $f(s,t)$ satisfies the curious recurrence

$$\begin{aligned} &f(s,0) = f(0,t) = 0; \qquad f(1,t) = t;\\ &f(s,t) = \tbinom{s+t-1}{s} + \max\bigl(f(t-1, s-1), f(s-2, t)\bigr) \quad \text{if } st > 0 \text{ and } s > 1. \end{aligned}$$

When $s + t = 2u + 2$ the solution turns out to be

$$f(s,t) = \binom{2u+1}{t-1} + \sum_{j=1}^{u-r} \binom{2u+1-2j}{r} + \sum_{j=0}^{r-1} \binom{2j+1}{j}, \qquad r = \min(s-2, t-1),$$

with the maximum occurring at $f(t-1, s-1)$ when $s \le t$ and at $f(s-2, t)$ when $s \ge t+2$.

Therefore the minimum $N - N'$ occurs for

$$\begin{aligned} C &= \{2m-1\} \cup \{2m-2-x \mid 1 \le x \le 2m-2,\ \ x \bmod 4 \le 1\},\\ C' &= \{2m-2\} \cup \{2m-2-x \mid 1 \le x \le 2m-2,\ \ x \bmod 4 \ge 2\}; \end{aligned}$$

and it equals $\binom{2m-1}{m-1} - \sum_{k=0}^{m-2} \binom{2k+1}{k} = 1 + \sum_{k=1}^{m-1} \binom{2k}{k-1}$. [See A. J. van Zanten, *IEEE Trans.* **IT-37** (1991), 1229–1233.]

26. (a) Yes: The first is $0^{n-\lceil t/2 \rceil} 1^{t \bmod 2} 2^{\lfloor t/2 \rfloor}$ and the last is $2^{\lfloor t/2 \rfloor} 1^{t \bmod 2} 0^{n - \lceil t/2 \rceil}$; transitions are substrings of the forms $02^a1 \leftrightarrow 12^a0$, $02^a2 \leftrightarrow 12^a1$, $10^a1 \leftrightarrow 20^a0$, $10^a2 \leftrightarrow 20^a1$.

(b) No: If $s = 0$ there is a big jump from 02^t0^{r-1} to 20^r2^{t-1}.

27. The following procedure extracts all combinations $c_1 \ldots c_k$ of Γ_n that have weight $\le t$: Begin with $k \leftarrow 0$ and $c_0 \leftarrow n$. Visit $c_1 \ldots c_k$. If k is even and $c_k = 0$, set $k \leftarrow k - 1$; if k is even and $c_k > 0$, set $c_k \leftarrow c_k - 1$ if $k = t$, otherwise $k \leftarrow k + 1$ and $c_k \leftarrow 0$. On the other hand if k is odd and $c_k + 1 = c_{k-1}$, set $k \leftarrow k - 1$ and

$c_k \leftarrow c_{k+1}$ (but terminate if $k = 0$); if k is odd and $c_k + 1 < c_{k-1}$, set $c_k \leftarrow c_k + 1$ if $k = t$, otherwise $k \leftarrow k + 1$, $c_k \leftarrow c_{k-1}$, $c_{k-1} \leftarrow c_k + 1$. Repeat.

(This loopless algorithm reduces to that of exercise 7.2.1.1–12(b) when $t = n$, with slight changes of notation.)

28. True. Bit strings $a_{n-1} \ldots a_0 = \alpha\beta$ and $a'_{n-1} \ldots a'_0 = \alpha\beta'$ correspond to index lists $(b_s \ldots b_1 = \theta\chi,\ c_t \ldots c_1 = \phi\psi)$ and $(b'_s \ldots b'_1 = \theta\chi',\ c'_t \ldots c'_1 = \phi\psi')$ such that everything between $\alpha\beta$ and $\alpha\beta'$ begins with α if and only if everything between $\theta\chi$ and $\theta\chi'$ begins with θ and everything between $\phi\psi$ and $\phi\psi'$ begins with ϕ. For example, if $n = 10$, the prefix $\alpha = 01101$ corresponds to prefixes $\theta = 96$ and $\phi = 875$.

(But just having $c_t \ldots c_1$ in genlex order is a much weaker condition. For example, *every* such sequence is genlex when $t = 1$.)

29. (a) $\texttt{-}^k\texttt{0}^{l+1}$ or $\texttt{-}^k\texttt{0}^{l+1}\texttt{+}\texttt{±}^m$ or $\texttt{±}^k$, for $k, l, m \ge 0$.

(b) No; the successor is always smaller in balanced ternary notation.

(c) For all α and all $k, l, m \ge 0$ we have $\alpha\texttt{0}\texttt{-}^{k+1}\texttt{0}^l\texttt{+}\texttt{±}^m \to \alpha\texttt{-}\texttt{+}^k\texttt{0}^{l+1}\texttt{-}\texttt{±}^m$ and $\alpha\texttt{+}\texttt{-}^k\texttt{0}^{l+1}\texttt{+}\texttt{±}^m \to \alpha\texttt{0}\texttt{+}^{k+1}\texttt{0}^l\texttt{-}\texttt{±}^m$; also $\alpha\texttt{0}\texttt{-}^{k+1}\texttt{0}^l \to \alpha\texttt{-}\texttt{+}^k\texttt{0}^{l+1}$ and $\alpha\texttt{+}\texttt{-}^k\texttt{0}^{l+1} \to \alpha\texttt{0}\texttt{+}^{k+1}\texttt{0}^l$.

(d) Let the jth sign of α_i be $(-1)^{a_{ij}}$, and let it be in position b_{ij}. Then we have $(-1)^{a_{ij}+b_{i(j-1)}} = (-1)^{a_{(i+1)j}+b_{(i+1)(j-1)}}$ for $0 \le i < k$ and $1 \le j \le t$, if we let $b_{i0} = 0$.

(e) By parts (a), (b), and (c), α belongs to some chain $\alpha_0 \to \cdots \to \alpha_k$, where α_k is final (has no successor) and α_0 is initial (has no predecessor). By part (d), every such chain has at most $\binom{s+t}{t}$ elements. But there are 2^s final strings, by (a), and there are $2^s\binom{s+t}{t}$ strings with s signs and t zeros; so k must be $\binom{s+t}{t} - 1$.

Reference: SICOMP **2** (1973), 128–133.

30. Assume that $t > 0$. Initial strings are the negatives of final strings. Let σ_j be the initial string $\texttt{0}^t\texttt{-}\tau_j$ for $0 \le j < 2^{s-1}$, where the kth character of τ_j for $1 \le k < s$ is the sign of $(-1)^{a_k}$ when j is the binary number $(a_{s-1} \ldots a_1)_2$; thus $\sigma_0 = \texttt{0}^t\texttt{-++}\ldots\texttt{+}$, $\sigma_1 = \texttt{0}^t\texttt{--+}\ldots\texttt{+}$, $\ldots$, $\sigma_{2^{s-1}-1} = \texttt{0}^t\texttt{---}\ldots\texttt{-}$. Let ρ_j be the final string obtained by inserting $\texttt{-0}^t$ after the first (possibly empty) run of minus signs in τ_j; thus $\rho_0 = \texttt{-0}^t\texttt{++}\ldots\texttt{+}$, $\rho_1 = \texttt{--0}^t\texttt{+}\ldots\texttt{+}$, $\ldots$, $\rho_{2^{s-1}-1} = \texttt{--}\ldots\texttt{-0}^t$. We also let $\sigma_{2^{s-1}} = \sigma_0$ and $\rho_{2^{s-1}} = \rho_0$. Then we can prove by induction that the chain beginning with σ_j ends with ρ_j when t is even, with ρ_{j-1} when t is odd, for $1 \le j \le 2^{s-1}$. Therefore the chain beginning with $-\rho_j$ ends with $-\sigma_j$ or $-\sigma_{j+1}$.

Let $A_j(s,t)$ be the sequence of (s,t)-combinations derived by mapping the chain that starts with σ_j, and let $B_j(s,t)$ be the analogous sequence derived from $-\rho_j$. Then, for $1 \le j \le 2^{s-1}$, the reverse sequence $A_j(s,t)^R$ is $B_j(s,t)$ when t is even, $B_{j-1}(s,t)$ when t is odd. The corresponding recurrences when $st > 0$ are

$$A_j(s,t) = \begin{cases} 1A_j(s,t-1),\ 0A_{\lfloor(2^{s-1}-1-j)/2\rfloor}(s-1,t)^R, & \text{if } j+t \text{ is even;} \\ 1A_j(s,t-1),\ 0A_{\lfloor j/2\rfloor}(s-1,t), & \text{if } j+t \text{ is odd;} \end{cases}$$

and when $st > 0$ all 2^{s-1} of these sequences are distinct.

Chase's sequence C_{st} is $A_{\lfloor 2^s/3\rfloor}(s,t)$, and $\widehat{C}_{st}$ is $A_{\lfloor 2^{s-1}/3\rfloor}(s,t)$. Incidentally, the homogeneous sequence K_{st} of (31) is $A_{2^{s-1}-[t\,\mathrm{even}]}(s,t)^R$.

31. (a) $2^{\binom{s+t}{t}-1}$ solves the recurrence $f(s,t) = 2f(s-1,t)f(s,t-1)$ when $f(s,0) = f(0,t) = 1$. (b) Now $f(s,t) = (s+1)!\,f(s,t-1)\ldots f(0,t-1)$ has the solution

$$(s+1)!^t\,s!^{\binom{t}{2}}(s-1)!^{\binom{t+1}{3}}\ldots 2!^{\binom{s+t-2}{s}} \;=\; \prod_{r=1}^{s}(r+1)!^{\binom{s+t-1-r}{t-2}+[r=s]}.$$

32. (a) No simple formula seems to exist, but the listings can be counted for small s and t by systematically computing the number of genlex paths that run through all weight-t strings from a given starting point to a given ending point via revolving-door moves. The totals for $s+t \le 6$ are

$$\begin{array}{ccccccccccccc}
&&&&&&1&&&&&&\\
&&&&&1&&1&&&&&\\
&&&&1&&2&&1&&&&\\
&&&1&&4&&4&&1&&&\\
&&1&&8&&20&&8&&1&&\\
&1&&16&&160&&160&&16&&1&\\
1&&32&&2264&&17152&&2264&&32&&1
\end{array}$$

and $f(4,4) = 95{,}304{,}112{,}865{,}280$; $f(5,5) \approx 5.92646 \times 10^{48}$. [This class of combination generators was first studied by G. Ehrlich, *JACM* **20** (1973), 500–513, but he did not attempt to enumerate them.]

(b) By extending the proof of Theorem N, one can show that all such listings or their reversals must run from 1^t0^s to $0^a1^t0^{s-a}$ for some a, $1 \le a \le s$. Moreover, the number n_{sta} of possibilities, given s, t, and a with $st > 0$, satisfies $n_{1t1} = 1$ and

$$n_{sta} = \begin{cases} n_{s(t-1)1}n_{(s-1)t(a-1)}, & \text{if } a > 1;\\ n_{s(t-1)2}n_{(s-1)t1} + \cdots + n_{s(t-1)s}n_{(s-1)t(s-1)}, & \text{if } a = 1 < s. \end{cases}$$

This recurrence has the remarkable solution $n_{sta} = 2^{m(s,t,a)}$, where

$$m(s,t,a) = \begin{cases} \binom{s+t-3}{t} + \binom{s+t-5}{t-2} + \cdots + \binom{s-1}{2}, & \text{if } t \text{ is even};\\ \binom{s+t-3}{t} + \binom{s+t-5}{t-2} + \cdots + \binom{s}{3} + s - a - [a<s], & \text{if } t \text{ is odd}. \end{cases}$$

33. Consider first the case $t = 1$: The number of near-perfect paths from i to $j > i$ is $f(j - i - [i>0] - [j<n-1])$, where $\sum_j f(j)z^j = 1/(1 - z - z^3)$. (By coincidence, the same sequence $f(j)$ arises in Caron's polyphase merge on 6 tapes, Table 5.4.2–2.) The sum over $0 \le i < j < n$ is $3f(n) + f(n-1) + f(n-2) + 2 - n$; and we must double this, to cover cases with $j > i$.

When $t > 1$ we can construct $\binom{n}{t} \times \binom{n}{t}$ matrices that tell how many genlex listings begin and end with particular combinations. The entries of these matrices are sums of products of matrices for the case $t-1$, summed over all paths of the type considered for $t = 1$. The totals for $s + t \le 6$ turn out to be

$$\begin{array}{ccccccccccccc}
&&&&&&1&&&&&&\\
&&&&&1&&1&&&&&\\
&&&&1&&2&&1&&&&\\
&&&1&&6&&2&&1&&&\\
&&1&&12&&10&&2&&1&&\\
&1&&20&&44&&10&&2&&1&\\
1&&34&&238&&68&&10&&2&&1
\end{array}
\qquad
\begin{array}{ccccccccccccc}
&&&&&&1&&&&&&\\
&&&&&1&&1&&&&&\\
&&&&1&&2&&1&&&&\\
&&&1&&2&&0&&1&&&\\
&&1&&2&&2&&0&&1&&\\
&1&&2&&0&&0&&0&&1&\\
1&&2&&6&&0&&0&&0&&1
\end{array}$$

where the right-hand triangle shows the number of *cycles*, $g(s,t)$. Further values include $f(4,4) = 17736$; $f(5,5) = 9{,}900{,}888{,}879{,}984$; $g(4,4) = 96$; $g(5,5) = 30{,}961{,}456{,}320$.

There are exactly 10 such schemes when $s = 2$ and $n \ge 4$. For example, when $n = 7$ they run from 43210 to 65431 or 65432, or from 54321 to 65420 or 65430 or 65432, or the reverse.

34. The minimum can be computed as in the previous answer, but using min-plus matrix multiplication $c_{ij} = \min_k(a_{ik} + b_{kj})$ instead of ordinary matrix multiplication $c_{ij} = \sum_k a_{ik}b_{kj}$. (When $s = t = 5$, the genlex path in Fig. 26(e) with only 49 imperfect transitions is essentially unique. There is a genlex cycle for $s = t = 5$ that has only 55 imperfections.)

35. From the recurrences (35) we have $a_{st} = b_{s(t-1)} + [s>1][t>0] + a_{(s-1)t}$, $b_{st} = a_{s(t-1)} + a_{(s-1)t}$; consequently $a_{st} = b_{st} + [s>1][t\,\text{odd}]$ and $a_{st} = a_{s(t-1)} + a_{(s-1)t} + [s>1][t\,\text{odd}]$. The solution is

$$a_{st} = \sum_{k=0}^{t/2} \binom{s+t-2-2k}{s-2} - [s>1][t\,\text{even}];$$

this sum is approximately $s/(s+2t)$ times $\binom{s+t}{t}$.

36. Consider the binary tree with root node (s,t) and with recursively defined subtrees rooted at $(s-1,t)$ and $(s,t-1)$ whenever $st>0$; the node (s,t) is a leaf if $st=0$. Then the subtree rooted at (s,t) has $\binom{s+t}{t}$ leaves, corresponding to all (s,t)-combinations $a_{n-1}\ldots a_1a_0$. Nodes on level l correspond to prefixes $a_{n-1}\ldots a_{n-l}$, and leaves on level l are combinations with $r = n-l$.

Any genlex algorithm for combinations $a_{n-1}\ldots a_1a_0$ corresponds to preorder traversal of such a tree, after the children of the $\binom{s+t}{t}-1$ branch nodes have been ordered in any desired way; that, in fact, is why there are $2^{\binom{s+t}{t}-1}$ such genlex schemes (exercise 31(a)). And the operation $j \leftarrow j+1$ is performed exactly once per branch node, namely after both children have been processed.

Incidentally, exercise 7.2.1.2–6(a) implies that the average value of r is $s/(t+1) + t/(s+1)$, which can be $\Omega(n)$; thus the extra time needed to keep track of r is worthwhile.

37. (a) In the lexicographic case we needn't maintain the w_j table, since a_j is active for $j \ge r$ if and only if $a_j = 0$. After setting $a_j \leftarrow 1$ and $a_{j-1} \leftarrow 0$ there are two cases to consider if $j>1$: If $r=j$, set $r \leftarrow j-1$; otherwise set $a_{j-2}\ldots a_0 \leftarrow 0^r1^{j-1-r}$ and $r \leftarrow j-1-r$ (or $r \leftarrow j$ if r was $j-1$).

(b) Now the transitions to be handled when $j>1$ are to change $a_j\ldots a_0$ as follows: $01^r \to 1101^{r-2}$, $010^r \to 10^{r+1}$, $010^a1^r \to 110^{a+1}1^{r-1}$, $10^r \to 010^{r-1}$, $110^r \to 010^{r-1}1$, $10^a1^r \to 0^a1^{r+1}$; these six cases are easily distinguished. The value of r should change appropriately.

(c) Again the case $j=1$ is trivial. Otherwise $01^a0^r \to 101^{a-1}0^r$; $0^a1^r \to 10^a1^{r-1}$; $101^a0^r \to 01^{a+1}0^r$; $10^a1^r \to 0^a1^{r+1}$; and there is also an ambiguous case, which can occur only if $a_{n-1}\ldots a_{j+1}$ contains at least one 0: Let $k>j$ be minimal with $a_k=0$. Then $10^r \to 010^{r-1}$ if k is odd, $10^r \to 0^r1$ if k is even.

38. The same algorithm works, except that (i) step C1 sets $a_{n-1}\ldots a_0 \leftarrow 01^t0^{s-1}$ if n is odd or $s=1$, $a_{n-1}\ldots a_0 \leftarrow 001^t0^{s-2}$ if n is even and $s>1$, with an appropriate value of r; (ii) step C3 interchanges the roles of even and odd; (iii) step C5 goes to C4 also if $j=1$.

39. In general, start with $r \leftarrow 0$, $j \leftarrow s+t-1$, and repeat the following steps until $st=0$:

$$r \leftarrow r + [w_j=0]\binom{j}{s-a_j}, \quad s \leftarrow s - [a_j=0], \quad t \leftarrow t - [a_j=1], \quad j \leftarrow j-1.$$

Then r is the rank of $a_{n-1}\ldots a_1a_0$. So the rank of 11001001000011111101101010 is $\binom{23}{12}+\binom{22}{11}+\binom{21}{9}+\binom{17}{8}+\binom{16}{7}+\binom{14}{5}+\binom{13}{3}+\binom{12}{3}+\binom{11}{3}+\binom{10}{3}+\binom{9}{3}+\binom{8}{3}+\binom{4}{3}+\binom{3}{1}+\binom{1}{0} = 2390131$.

40. We start with $N \leftarrow 999999$, $v \leftarrow 0$, and repeat the following steps until $st = 0$: If $v = 0$, set $t \leftarrow t-1$ and $a_{s+t} \leftarrow 1$ if $N < \binom{s+t-1}{s}$, otherwise set $N \leftarrow N - \binom{s+t-1}{s}$, $v \leftarrow (s+t) \bmod 2$, $s \leftarrow s-1$, $a_{s+t} \leftarrow 0$. If $v = 1$, set $v \leftarrow (s+t) \bmod 2$, $s \leftarrow s-1$, and $a_{s+t} \leftarrow 0$ if $N < \binom{s+t-1}{t}$, otherwise set $N \leftarrow N - \binom{s+t-1}{t}$, $t \leftarrow t-1$, $a_{s+t} \leftarrow 1$. Finally if $s = 0$, set $a_{t-1}\ldots a_0 \leftarrow 1^t$; if $t = 0$, set $a_{s-1}\ldots a_0 \leftarrow 0^s$. The answer is $a_{25}\ldots a_0 = 11101001111110101001000001$.

41. Let $c(0), \ldots, c(2^n-1) = C_n$ where $C_{2n} = 0C_{2n-1}, 1C_{2n-1}$; $C_{2n+1} = 0C_{2n}, 1\widehat{C}_{2n}$; $\widehat{C}_{2n} = 1C_{2n-1}, 0\widehat{C}_{2n-1}$; $\widehat{C}_{2n+1} = 1\widehat{C}_{2n}, 0\widehat{C}_{2n}$; $C_0 = \widehat{C}_0 = \epsilon$. Then $a_j \oplus b_j = b_{j+1} \wedge (b_{j+2} \vee (b_{j+3} \wedge (b_{j+4} \vee \cdots)))$ if j is even, $b_{j+1} \vee (b_{j+2} \wedge (b_{j+3} \vee (b_{j+4} \wedge \cdots)))$ if j is odd. Curiously we also have the inverse relation $c((\ldots a_4\bar{a}_3a_2\bar{a}_1a_0)_2) = (\ldots b_4\bar{b}_3b_2\bar{b}_1b_0)_2$.

42. Equation (40) shows that the left context $a_{n-1}\ldots a_{l+1}$ does not affect the behavior of the algorithm on $a_{l-1}\ldots a_0$ if $a_l = 0$ and $l > r$. Therefore we can analyze Algorithm C by counting combinations that end with certain bit patterns, and it follows that the number of times each operation is performed can be represented as $[w^s z^t]\, p(w,z)/(1-w^2)^2(1-z^2)^2(1-w-z)$ for an appropriate polynomial $p(w,z)$.

For example, the algorithm goes from C5 to C4 once for each combination that ends with $01^{2a+1}01^{2b+1}$ or has the form $1^{a+1}01^{2b+1}$, for integers $a, b \ge 0$; the corresponding generating functions are $w^2z^2/(1-z^2)^2(1-w-z)$ and $w(z^2+z^3)/(1-z^2)^2$.

Here are the polynomials $p(w,z)$ for key operations. Let $W = 1-w^2$, $Z = 1-z^2$.

C3 → C4:	$wzW(1+wz)(1-w-z^2)$;	C5($r \leftarrow 1$):	$w^2zW^2Z(1-wz-z^2)$;
C3 → C5:	$wzW(w+z)(1-wz-z^2)$;	C5($r \leftarrow j-1$):	$w^2z^3W^2(1-wz-z^2)$;
C3 → C6:	$w^2z^2W(w+z)$;	C6($j = 1$):	w^2zW^2Z;
C3 → C7:	$w^2zW(1+wz)$;	C6($r \leftarrow j-1$):	$w^2z^3W^2$;
C4($j = 1$):	$wzW^2Z(1-w-z^2)$;	C6($r \leftarrow j$):	w^3z^2WZ;
C4($r \leftarrow j-1$):	$w^3zWZ(1-w-z^2)$;	C7 → C6:	w^2zW^2;
C4($r \leftarrow j$):	$wz^2W^2(1+z-2wz-z^2-z^3)$;	C7($r \leftarrow j$):	w^4zWZ;
C5 → C4:	$wz^2W^2(1-wz-z^2)$;	C7($r \leftarrow j-2$):	$w^3z^2W^2$.
C5($r \leftarrow j-2$):	$w^4zWZ(1-wz-z^2)$;		

The asymptotic value is $\binom{s+t}{t}\bigl(p(1-x,x)/(2x-x^2)^2(1-x^2)^2 + O(n^{-1})\bigr)$, for fixed $0 < x < 1$, if $t = xn + O(1)$ as $n \to \infty$. Thus we find, for example, that the four-way branching in step C3 takes place with relative frequencies $x + x^2 - x^3 : 1 : x : 1 + x - x^2$.

Incidentally, the number of cases with j odd exceeds the number of cases with j even by

$$\sum_{k,l\ge 1} \binom{s+t-2k-2l}{s-2k} [2k+2l \le s+t] + [s\,\mathrm{odd}][t\,\mathrm{odd}],$$

in *any* genlex scheme that uses (39). This quantity has the interesting generating function $wz/(1+w)(1+z)(1-w-z)$.

43. The identity is true for all nonnegative integers x, except when $x = 1$.

44. In fact, $C_t(n) - 1 = \widehat{C}_t(n-1)^R$, and $\widehat{C}_t(n) - 1 = C_t(n-1)^R$. (Hence $C_t(n) - 2 = C_t(n-2)$, etc.)

45. In the following algorithm, r is the least subscript with $c_r \ge r$.

CC1. [Initialize.] Set $c_j \leftarrow n - t - 1 + j$ and $z_j \leftarrow 0$ for $1 \le j \le t+1$. Also set $r \leftarrow 1$. (We assume that $0 < t < n$.)

CC2. [Visit.] Visit the combination $c_t \ldots c_2c_1$. Then set $j \leftarrow r$.

CC3. [Branch.] Go to CC5 if $z_j \ne 0$.

CC4. [Try to decrease c_j.] Set $x \leftarrow c_j + (c_j \bmod 2) - 2$. If $x \geq j$, set $c_j \leftarrow x$, $r \leftarrow 1$; otherwise if $c_j = j$, set $c_j \leftarrow j-1$, $z_j \leftarrow c_{j+1} - \big((c_{j+1}+1) \bmod 2\big)$, $r \leftarrow j$; otherwise if $c_j < j$, set $c_j \leftarrow j$, $z_j \leftarrow c_{j+1} - \big((c_{j+1}+1) \bmod 2\big)$, $r \leftarrow \max(1, j-1)$; otherwise set $c_j \leftarrow x$, $r \leftarrow j$. Return to CC2.

CC5. [Try to increase c_j.] Set $x \leftarrow c_j + 2$. If $x < z_j$, set $c_j \leftarrow x$; otherwise if $x = z_j$ and $z_{j+1} \neq 0$, set $c_j \leftarrow x - (c_{j+1} \bmod 2)$; otherwise set $z_j \leftarrow 0$, $j \leftarrow j+1$, and go to CC3 (but terminate if $j > t$). If $c_1 > 0$, set $r \leftarrow 1$; otherwise set $r \leftarrow j-1$. Return to CC2. ▌

46. Equation (40) implies that $u_k = (b_j + k + 1) \bmod 2$ when j is minimal with $b_j > k$. Then (37) and (38) yield the following algorithm, where we assume for convenience that $3 \leq s < n$.

CB1. [Initialize.] Set $b_j \leftarrow j-1$ for $1 \leq j \leq s$; also set $z \leftarrow s+1$, $b_z \leftarrow 1$. (When subsequent steps examine the value of z, it is the smallest index such that $b_z \neq z-1$.)

CB2. [Visit.] Visit the dual combination $b_s \ldots b_2 b_1$.

CB3. [Branch.] If b_2 is odd: Go to CB4 if $b_2 \neq b_1+1$, otherwise to CB5 if $b_1 > 0$, otherwise to CB6 if b_z is odd. Go to CB9 if b_2 is even and $b_1 > 0$. Otherwise go to CB8 if $b_{z+1} = b_z + 1$, otherwise to CB7.

CB4. [Increase b_1.] Set $b_1 \leftarrow b_1 + 1$ and return to CB2.

CB5. [Slide b_1 and b_2.] If b_3 is odd, set $b_1 \leftarrow b_1 + 1$ and $b_2 \leftarrow b_2 + 1$; otherwise set $b_1 \leftarrow b_1 - 1$, $b_2 \leftarrow b_2 - 1$, $z \leftarrow 3$. Go to CB2.

CB6. [Slide left.] If z is odd, set $z \leftarrow z-2$, $b_{z+1} \leftarrow z+1$, $b_z \leftarrow z$; otherwise set $z \leftarrow z-1$, $b_z \leftarrow z$. Go to CB2.

CB7. [Slide b_z.] If b_{z+1} is odd, set $b_z \leftarrow b_z + 1$ and terminate if $b_z \geq n$; otherwise set $b_z \leftarrow b_z - 1$, then if $b_z < z$ set $z \leftarrow z+1$. To CB2.

CB8. [Slide b_z and b_{z+1}.] If b_{z+2} is odd, set $b_z \leftarrow b_{z+1}$, $b_{z+1} \leftarrow b_z + 1$, and terminate if $b_{z+1} \geq n$. Otherwise set $b_{z+1} \leftarrow b_z$, $b_z \leftarrow b_z - 1$, then if $b_z < z$ set $z \leftarrow z+2$. To CB2.

CB9. [Decrease b_1.] Set $b_1 \leftarrow b_1 - 1$, $z \leftarrow 2$, and return to CB2. ▌

Notice that this algorithm is *loopless*. Chase gave a similar procedure for the sequence $\widehat{C}^R_{st}$ in *Cong. Num.* **69** (1989), 233–237. It is truly amazing that this algorithm defines precisely the complements of the indices $c_t \ldots c_1$ produced by the algorithm in the previous exercise.

47. We can, for example, use Algorithm C and its reverse (exercise 38), with w_j replaced by a d-bit number whose bits represent activity at different levels of the recursion. Separate pointers $r_0, r_1, \ldots, r_{d-1}$ are needed to keep track of the r-values on each level. (Many other solutions are possible.)

48. There are permutations $\pi_1, \ldots, \pi_M$ such that the kth element of Λ_j is $\pi_k \alpha_j {\uparrow} \beta_{k-1}$. And $\pi_k \alpha_j$ runs through all permutations of $\{s_1 \cdot 1, \ldots, s_d \cdot d\}$ as j varies from 0 to $N-1$.

Historical note: The first publication of a homogeneous revolving-door scheme for (s,t)-combinations was by Éva Török, *Matematikai Lapok* **19** (1968), 143–146, who was motivated by the generation of multiset permutations. Many authors have subsequently relied on the homogeneity condition for similar constructions, but this exercise shows that homogeneity is not necessary.

49. We have $\lim_{z\to q}(z^{km+r}-1)/(z^{lm+r}-1) = 1$ when $0 < r < m$, and the limit is $\lim_{z\to q}(kmz^{km-1})/(lmz^{lm-1}) = k/l$ when $r = 0$. So we can pair up factors of the numerator $\prod_{n-k<a\le n}(z^a-1)$ with factors of the denominator $\prod_{0<b\le k}(z^b-1)$ when $a \equiv b$ (modulo m).

Notes: This formula was discovered by G. Olive, *AMM* **72** (1965), 619. In the special case $m = 2$, $q = -1$, the second factor vanishes only when n is even and k is odd. The formula $\binom{n}{k}_q = \binom{n}{n-k}_q$ holds for all $n \ge 0$, but $\binom{\lfloor n/m\rfloor}{\lfloor k/m\rfloor}$ is *not* always equal to $\binom{\lfloor n/m\rfloor}{\lfloor (n-k)/m\rfloor}$. We do, however, have $\lfloor k/m\rfloor + \lfloor (n-k)/m\rfloor = \lfloor n/m\rfloor$ in the case when $n \bmod m \ge k \bmod m$; otherwise the second factor is zero.

50. The stated coefficient is zero when $n_1 \bmod m + \cdots + n_t \bmod m \ge m$. Otherwise it equals

$$\binom{\lfloor (n_1+\cdots+n_t)/m\rfloor}{\lfloor n_1/m\rfloor,\ldots,\lfloor n_t/m\rfloor}\binom{(n_1+\cdots+n_t)\bmod m}{n_1 \bmod m,\ldots,n_t \bmod m}_q,$$

by Eq. 1.2.6–(43); here each upper index is the sum of the lower indices.

51. All paths clearly run between 000111 and 111000, since those vertices have degree 1. Fourteen total paths reduce to four under the stated equivalences. The path in (50), which is equivalent to itself under reflection-and-reversal, can be described by the delta sequence A = 3452132523414354123; the other three classes are B = 3452541453414512543, C = 3452541453252154123, D = 3452134145341432543. D. H. Lehmer found path C [*AMM* **72** (1965), Part II, 36–46]; D is essentially the path constructed by Eades, Hickey, and Read.

(Incidentally, perfect schemes aren't really rare, although they seem to be difficult to construct systematically. The case $(s,t) = (3,5)$ has 4,050,046 of them.)

52. We may assume that each s_j is nonzero and that $d > 1$. Then the difference between permutations with an even and odd number of inversions is $\binom{\lfloor (s_0+\cdots+s_d)/2\rfloor}{\lfloor s_0/2\rfloor,\ldots,\lfloor s_d/2\rfloor} \ge 2$, by exercise 50, unless at least two of the multiplicities s_j are odd.

Conversely, if at least two multiplicities are odd, a general construction by G. Stachowiak [*SIAM J. Discrete Math.* **5** (1992), 199–206] shows that a perfect scheme exists. Indeed, his construction applies to a variety of topological sorting problems; in the special case of multisets it gives a Hamiltonian cycle in all cases with $d > 1$ and s_0s_1 odd, except when $d = 2$, $s_0 = s_1 = 1$, and s_2 is even.

53. See *AMM* **72** (1965), Part II, 36–46.

54. Assuming that $st \ne 0$, a Hamiltonian path exists if and only if s and t are not both even; a Hamiltonian cycle exists if and only if, in addition, ($s \ne 2$ and $t \ne 2$) or $n = 5$. [T. C. Enns, *Discrete Math.* **122** (1993), 153–165.]

55. [Solution by Aaron Williams.] The sequence 0^s1^t, W_{st} has the correct properties if

$$W_{st} \;=\; 0W_{(s-1)t},\ 1W_{s(t-1)},\ 10^s1^{t-1}, \quad \text{for } st > 0; \qquad W_{0t} = W_{s0} = \emptyset.$$

And there is an amazingly efficient, *loopless* implementation: Assume that $t > 0$.

W1. [Initialize.] Set $n \leftarrow s+t$, $a_j \leftarrow 1$ for $0 \le j < t$, and $a_j \leftarrow 0$ for $t \le i \le n$. Also set $j \leftarrow k \leftarrow t-1$. (This is tricky, but it works.)

W2. [Visit.] Visit the (s,t)-combination $a_{n-1}\ldots a_1a_0$.

W3. [Zero out a_j.] Set $a_j \leftarrow 0$ and $j \leftarrow j+1$.

W4. [Easy case?] If $a_j = 1$, set $a_k \leftarrow 1$, $k \leftarrow k+1$, and return to W2.

W5. [Wrap around.] Terminate if $j = n$. Otherwise set $a_j \leftarrow 1$. Then if $k > 0$, set $a_k \leftarrow 1$, $a_0 \leftarrow 0$, $j \leftarrow 1$, and $k \leftarrow 0$. Return to W2. ▮

After the second visit, j is the smallest index with $a_j a_{j-1} = 10$, and k is smallest with $a_k = 0$. The easy case occurs exactly $\binom{s+t-1}{s} - 1$ times; and the condition $k = 0$ occurs in step W5 exactly $\binom{s+t-2}{t} + \delta_{t1}$ times. [To appear.]

56. [*Discrete Math.* **48** (1984), 163–171.] This problem is equivalent to the "middle levels conjecture," which states that there is a Gray path through all binary strings of length $2t-1$ and weights $\{t-1, t\}$. In fact, such strings can almost certainly be generated by a delta sequence of the special form $\alpha_0\alpha_1 \ldots \alpha_{2t-2}$ where the elements of α_k are those of α_0 shifted by k, modulo $2t-1$. For example, when $t = 3$ we can start with $a_5a_4a_3a_2a_1a_0 = 000111$ and repeatedly swap $a_0 \leftrightarrow a_\delta$, where δ runs through the cycle (4134 5245 1351 2412 3523). The middle levels conjecture is known to be true for $t \le 15$ [see I. Shields and C. D. Savage, *Cong. Num.* **140** (1999), 161–178].

57. Yes; there is a near-perfect genlex solution for all m, n, and t when $n \ge m > t$. One such scheme, in bitstring notation, is $1A_{(m-t)(t-1)}0^{n-m}$, $01A_{(m-t)(t-1)}0^{n-m-1}$, $\ldots$, $0^{n-m}1A_{(m-t)(t-1)}$, $0^{n-m+1}1A_{(m-1-t)(t-1)}$, $\ldots$, $0^{n-t}1A_{0(t-1)}$, using the sequences A_{st} of (35).

58. Solve the previous problem with m and n reduced by $t-1$, then add $j-1$ to each c_j. (Case (a), which is particularly simple, was probably known to Czerny.)

59. The generating function $G_{mnt}(z) = \sum g_{mntk} z^k$ for the number g_{mntk} of chords reachable in k steps from $0^{n-t}1^t$ satisfies $G_{mmt}(z) = \binom{m}{t}_z$ and $G_{m(n+1)t}(z) = G_{mnt}(z) + z^{tn-(t-1)m}\binom{m-1}{t-1}_z$, because the latter term accounts for cases with $c_t = n$ and $c_1 > n - m$. A perfect scheme is possible only if $|G_{mnt}(-1)| \le 1$. But if $n \ge m > t \ge 2$, this condition holds only when $m = t+1$ or $(n-t)t$ is odd, by (49). So there is no perfect solution when $t = 4$ and $m > 5$. (Many chords have only two neighbors when $n = t+2$, so one can easily rule out that case. All cases with $n \ge m > 5$ and $t = 3$ apparently do have perfect paths when n is even.)

60. The following solution uses lexicographic order, taking care to ensure that the average amount of computation per visit is bounded. We may assume that $stm_s \ldots m_0 \ne 0$ and $t \le m_s + \cdots + m_1 + m_0$.

Q1. [Initialize.] Set $q_j \leftarrow 0$ for $s \ge j \ge 1$, and $x = t$.

Q2. [Distribute.] Set $j \leftarrow 0$. Then while $x > m_j$, set $q_j \leftarrow m_j$, $x \leftarrow x - m_j$, $j \leftarrow j+1$, and repeat until $x \le m_j$. Finally set $q_j \leftarrow x$.

Q3. [Visit.] Visit the bounded composition $q_s + \cdots + q_1 + q_0$.

Q4. [Pick up the rightmost units.] If $j = 0$, set $x \leftarrow q_0 - 1$, $j \leftarrow 1$. Otherwise if $q_0 = 0$, set $x \leftarrow q_j - 1$, $q_j \leftarrow 0$, and $j \leftarrow j+1$. Otherwise go to Q7.

Q5. [Full?] Terminate if $j > s$. Otherwise if $q_j = m_j$, set $x \leftarrow x + m_j$, $q_j \leftarrow 0$, $j \leftarrow j+1$, and repeat this step.

Q6. [Increase q_j.] Set $q_j \leftarrow q_j + 1$. Then if $x = 0$, set $q_0 \leftarrow 0$ and return to Q3. (In that case $q_{j-1} = \cdots = q_0 = 0$.) Otherwise go to Q2.

Q7. [Increase and decrease.] (Now $q_i = m_i$ for $j > i \ge 0$.) While $q_j = m_j$, set $j \leftarrow j+1$ and repeat until $q_j < m_j$ (but terminate if $j > s$). Then set $q_j \leftarrow q_j + 1$, $j \leftarrow j-1$, $q_j \leftarrow q_j - 1$. If $q_0 = 0$, set $j \leftarrow 1$. Return to Q3. ▌

For example, if $m_s = \cdots = m_0 = 9$, the successors of the composition $3{+}9{+}9{+}7{+}0{+}0$ are $4{+}0{+}0{+}6{+}9{+}9$, $4{+}0{+}0{+}7{+}8{+}9$, $4{+}0{+}0{+}7{+}9{+}8$, $4{+}0{+}0{+}8{+}7{+}9$, $\ldots$.

61. Let $F_s(t) = \emptyset$ if $t < 0$ or $t > m_s + \cdots + m_0$; otherwise let $F_0(t) = t$, and

$$F_s(t) = 0{+}F_{s-1}(t),\ 1{+}F_{s-1}(t-1)^R,\ 2{+}F_{s-1}(t-2),\ \ldots,\ m_s{+}F_{s-1}(t-m_s)^{R^{m_s}}$$

when $s > 0$. This sequence can be shown to have the required properties; it is, in fact, equivalent to the compositions defined by the homogeneous sequence K_{st} of (31) under the correspondence of exercise 4, when restricted to the subsequence defined by the bounds m_s, …, m_0. [See T. Walsh, *J. Combinatorial Math. and Combinatorial Computing* **33** (2000), 323–345, who has implemented it looplessly.]

62. (a) A $2 \times n$ contingency table with row sums r and $c_1 + \cdots + c_n - r$ is equivalent to solving $r = a_1 + \cdots + a_n$ with $0 \le a_1 \le c_1$, …, $0 \le a_n \le c_n$.

(b) We can compute it sequentially by setting $a_{ij} \leftarrow \min(r_i - a_{i1} - \cdots - a_{i(j-1)}, c_j - a_{1j} - \cdots - a_{(i-1)j})$ for $j = 1, \ldots, n$, for $i = 1, \ldots, m$. Alternatively, if $r_1 \le c_1$, set $a_{11} \leftarrow r_1$, $a_{12} \leftarrow \cdots \leftarrow a_{1n} \leftarrow 0$, and do the remaining rows with c_1 decreased by r_1; if $r_1 > c_1$, set $a_{11} \leftarrow c_1$, $a_{21} \leftarrow \cdots \leftarrow a_{m1} \leftarrow 0$, and do the remaining columns with r_1 decreased by c_1. The second approach shows that at most $m + n - 1$ of the entries are nonzero. We can also write down the explicit formula

$$a_{ij} = \max(0, \min(r_i, c_j, r_1 + \cdots + r_i - c_1 - \cdots - c_{j-1}, c_1 + \cdots + c_j - r_1 - \cdots - r_{i-1})).$$

(c) The same matrix is obtained as in (b).

(d) Reverse left and right in (b) and (c); in both cases the answer is

$$a_{ij} = \max(0, \min(r_i, c_j, r_{i+1} + \cdots + r_m - c_1 - \cdots - c_{j-1}, c_1 + \cdots + c_j - r_i - \cdots - r_m)).$$

(e) Here we choose, say, row-wise order: Generate the first row just as for bounded compositions of r_1, with bounds $(c_1, \ldots, c_n)$; and for each row $(a_{11}, \ldots, a_{1n})$, generate the remaining rows recursively in the same way, but with the column sums $(c_1 - a_{11}, \ldots, c_n - a_{1n})$. Most of the action takes place on the bottom two rows, but when a change is made to an earlier row the later rows must be re-initialized.

63. If a_{ij} and a_{kl} are positive, we obtain another contingency table by setting $a_{ij} \leftarrow a_{ij} - 1$, $a_{il} \leftarrow a_{il} + 1$, $a_{kj} \leftarrow a_{kj} + 1$, $a_{kl} \leftarrow a_{kl} - 1$. We want to show that the graph G whose vertices are the contingency tables for $(r_1, \ldots, r_m; c_1, \ldots, c_n)$, adjacent if they can be obtained from each other by such a transformation, has a Hamiltonian path.

When $m = n = 2$, G is a simple path. When $m = 2$ and $n = 3$, G has a two-dimensional structure from which we can see that every vertex is the starting point of at least two Hamiltonian paths, having distinct endpoints. When $m = 2$ and $n \ge 4$ we can show, inductively, that G actually has Hamiltonian paths from any vertex to any other.

When $m \ge 3$ and $n \ge 3$, we can reduce the problem from m to $m - 1$ as in answer 62(e), if we are careful not to "paint ourselves into a corner." Namely, we must avoid reaching a state where the nonzero entries of the bottom two rows have the form $\binom{1\ a\ 0}{0\ b\ c}$ for some a, b, $c > 0$ and a change to row $m - 2$ forces this to become $\binom{0\ a\ 1}{0\ b\ c}$. The previous round of changes to rows $m - 1$ and m can avoid such a trap unless $c = 1$ and it begins with $\binom{0\ a{+}1\ 0}{1\ b{-}1\ 1}$ or $\binom{1\ a{-}1\ 1}{0\ b{+}1\ 0}$. But that situation can be avoided too.

(A genlex method based on exercise 61 would be considerably simpler, and it almost always would make only four changes per step. But it would occasionally need to update $2\min(m, n)$ entries at a time.)

64. When $x_1 \ldots x_s$ is a binary string and A is a list of subcubes, let $A \oplus x_1 \ldots x_s$ denote replacing the digits $(a_1, \ldots, a_s)$ in each subcube of A by $(a_1 \oplus x_1, \ldots, a_s \oplus x_s)$, from left to right. For example, $0{*}1{*}{*}10 \oplus 1010 = 1{*}1{*}{*}00$. Then the following mutual recursions define a Gray cycle, because A_{st} gives a Gray path from $0^s{*}^t$ to $10^{s-1}{*}^t$ and B_{st} gives a Gray path from $0^s{*}^t$ to ${*}01^{s-1}{*}^{t-1}$, when $st > 0$:

$$A_{st} \;=\; 0B_{(s-1)t},\ {*}A_{s(t-1)} \oplus 001^{s-2},\ 1B^R_{(s-1)t};$$

$$B_{st} \;=\; 0A_{(s-1)t},\ 1B_{(s-1)t}\oplus 010^{s-2},\ {*}A_{s(t-1)}\oplus 1^s.$$

The strings 001^{s-2} and 010^{s-2} are simply 0^s when $s<2$; A_{s0} is Gray binary code; $A_{0t}=B_{0t}={*}^t$. (Incidentally, the somewhat simpler construction

$$G_{st} \;=\; {*}G_{s(t-1)},\ a_tG_{(s-1)t},\ a_{t-1}G^R_{(s-1)t}, \qquad a_t = t \bmod 2,$$

defines a pleasant Gray *path* from ${*}^t0^s$ to $a_{t-1}{*}^t0^{s-1}$.)

65. If a path P is considered equivalent to P^R and to $P\oplus x_1\ldots x_s$, the total number can be computed systematically as in exercise 33, with the following results for $s{+}t\le 6$:

paths	cycles
1	1
1 1	1 1
1 2 1	1 1 1
1 3 3 1	1 1 1 1
1 5 10 4 1	1 2 1 1 1
1 6 36 35 5 1	1 2 3 1 1 1
1 9 310 4630 218 6 1	1 3 46 4 1 1 1

In general there are $t+1$ paths when $s=1$ and $\binom{\lceil s/2\rceil+2}{2}-(s \bmod 2)$ when $t=1$. The cycles for $s\le 2$ are unique. When $s=t=5$ there are approximately 6.869×10^{170} paths and 2.495×10^{70} cycles.

66. Let $G(n,0)=\epsilon$; $G(n,t)=\emptyset$ when $n<t$; and for $1\le t\le n$, let $G(n,t)$ be

$$\hat g(0)G(n-1,t),\ \hat g(1)G(n-1,t)^R,\ \ldots,\ \hat g(2^t-1)G(n-1,t)^R,\ \hat g(2^t-1)G(n-1,t-1),$$

where $\hat g(k)$ is a t-bit column containing the Gray binary number $g(k)$ with its least significant bit at the top. In this general formula we implicitly add a row of zeros below the bases of $G(n-1,t-1)$.

This remarkable rule gives ordinary Gray binary code when $t=1$, omitting $0\ldots00$. A cyclic Gray code is impossible because $\binom{n}{t}_2$ is odd.

67. A Gray path for compositions corresponding to Algorithm C implies that there is a path in which all transitions are $0^k1^l\leftrightarrow 1^l0^k$ with $\min(k,l)\le 2$. Perhaps there is, in fact, a cycle with $\min(k,l)=1$ in each transition.

68. (a) $\{\emptyset\}$; (b) $\emptyset$.

69. The least N with $\kappa_tN<N$ is $\binom{2t-1}{t}+\binom{2t-3}{t-1}+\cdots+\binom{1}{1}+1=\frac12\bigl(\binom{2t}{t}+\binom{2t-2}{t-1}+\cdots+\binom{0}{0}+1\bigr)$, because $\binom{n}{t-1}\le\binom{n}{t}$ if and only if $n\ge 2t-1$.

70. From the identity

$$\kappa_t\bigl(\tbinom{2t-3}{t}+N'\bigr)-\bigl(\tbinom{2t-3}{t}+N'\bigr)=\kappa_t\bigl(\tbinom{2t-2}{t}+N'\bigr)-\bigl(\tbinom{2t-2}{t}+N'\bigr)=\tbinom{2t-2}{t}\tfrac{1}{t-1}+\kappa_{t-1}N'-N'$$

when $N'<\binom{2t-3}{t}$, we conclude that the maximum is $\binom{2t-2}{t}\frac1t+\binom{2t-4}{t-1}\frac{1}{t-2}+\cdots+\binom{2}{2}\frac11$, and it occurs at 2^{t-1} values of N when $t>1$.

71. Let C_t be the t-cliques. The first $\binom{1414}{t}+\binom{1009}{t-1}$ t-combinations visited by Algorithm L define a graph on 1415 vertices with 1000000 edges. If $|C_t|$ were larger, $|\partial^{t-2}C_t|$ would exceed 1000000. Thus the single graph defined by $P_{(1000000)2}$ has the maximum number of t-cliques for all $t\ge 2$.

72. $M=\binom{m_s}{s}+\cdots+\binom{m_u}{u}$ for $m_s>\cdots>m_u\ge u\ge 1$, where $\{m_s,\ldots,m_u\}=\{s{+}t{-}1,\ldots,n_v\}\backslash\{n_t,\ldots,n_{v+1}\}$. (Compare with exercise 15, which gives $\binom{s+t}{t}-1-N$.)

If $\alpha = a_{n-1} \ldots a_0$ is the bit string corresponding to the combination $n_t \ldots n_1$, then v is 1 plus the number of trailing 1s in α, and u is the length of the rightmost run of 0s. For example, when $\alpha = 1010001111$ we have $s = 4$, $t = 6$, $M = \binom{8}{4} + \binom{7}{3}$, $u = 3$, $N = \binom{9}{6} + \binom{7}{5}$, $v = 5$.

73. A and B are cross-intersecting $\iff \alpha \not\subseteq U \setminus \beta$ for all $\alpha \in A$ and $\beta \in B \iff A \cap \partial^{n-s-t} B^- = \emptyset$, where $B^- = \{U \setminus \beta \mid \beta \in B\}$ is a set of $(n-t)$-combinations. Since $Q^-_{Nnt} = P_{N(n-t)}$, we have $|\partial^{n-s-t} B^-| \geq |\partial^{n-s-t} P_{N(n-t)}|$, and $\partial^{n-s-t} P_{N(n-t)} = P_{N's}$ where $N' = \kappa_{s+1} \ldots \kappa_{n-t} N$. Thus if A and B are cross-intersecting we have $M + N' \leq |A| + |\partial^{n-s-t} B^-| \leq \binom{n}{s}$, and $Q_{Mns} \cap P_{N's} = \emptyset$.

Conversely, if $Q_{Mns} \cap P_{N's} \neq \emptyset$ we have $\binom{n}{s} < M + N' \leq |A| + |\partial^{n-s-t} B^-|$, so A and B cannot be cross-intersecting.

74. $|\varrho Q_{Nnt}| = \kappa_{n-t} N$ (see exercise 94). Also, arguing as in (58) and (59), we find $\varrho P_{N5} = (n-1) P_{N5} \cup \cdots \cup 10 P_{N5} \cup \{543210, \ldots, 987654\}$ in that particular case; and $|\varrho P_{Nt}| = (n + 1 - n_t) N + \binom{n_t+1}{t+1}$ in general.

75. The identity $\binom{n+1}{k} = \binom{n}{k} + \binom{n-1}{k-1} + \cdots + \binom{n-k}{0}$, Eq. 1.2.6–(10), gives another representation if $n_v > v$. But (60) is unaffected, since we have $\binom{n+1}{k-1} = \binom{n}{k-1} + \binom{n-1}{k-2} + \cdots + \binom{n-k+1}{0}$.

76. Represent $N+1$ by adding $\binom{v-1}{v-1}$ to (57); then use the previous exercise to deduce that $\kappa_t(N+1) - \kappa_t N = \binom{v-1}{v-2} = v - 1$.

77. [D. E. Daykin, *Nanta Math.* **8**, 2 (1975), 78–83.] We work with extended representations $M = \binom{m_t}{t} + \cdots + \binom{m_u}{u}$ and $N = \binom{n_t}{t} + \cdots + \binom{n_v}{v}$ as in exercise 75, calling them *improper* if the final index u or v is zero. Call N *flexible* if it has both proper and improper representations, that is, if $n_v > v > 0$.

(a) Given an integer S, find $M + N$ such that $M + N = S$ and $\kappa_t M + \kappa_t N$ is minimum, with M as large as possible. If $N = 0$, we're done. Otherwise the max-min operation preserves both $M + N$ and $\kappa_t M + \kappa_t N$, so we can assume that $v \geq u \geq 1$ in the proper representations of M and N. If N is inflexible, $\kappa_t(M+1) + \kappa_t(N-1) = (\kappa_t M + u - 1) + (\kappa_t N - v) < \kappa_t M + \kappa_t N$, by exercise 76; therefore N must be flexible. But then we can apply the max-min operation to M and the improper representation of N, increasing M: Contradiction.

This proof shows that equality holds if and only if $MN = 0$, a fact that was noted in 1927 by F. S. Macaulay.

(b) Now we try to minimize $\max(\kappa_t M, N) + \kappa_{t-1} N$ when $M + N = S$, this time representing N as $\binom{n_{t-1}}{t-1} + \cdots + \binom{n_v}{v}$. The max-min operation can still be used if $n_{t-1} < m_t$; leaving m_t unchanged, it preserves $M + N$ and $\kappa_t M + \kappa_{t-1} N$ as well as the relation $\kappa_t M > N$. We arrive at a contradiction as in (a) if $N \neq 0$, so we can assume that $n_{t-1} \geq m_t$.

If $n_{t-1} > m_t$ we have $N > \kappa_t M$ and also $\lambda_t N > M$; hence $M + N < \lambda_t N + N = \binom{n_{t-1}+1}{t} + \cdots + \binom{n_v+1}{v}$, and we have $\kappa_t(M + N) \leq \kappa_t(\lambda_t N + N) = N + \kappa_{t-1} N$.

Finally if $n_{t-1} = m_t = a$, let $M = \binom{a}{t} + M'$ and $N = \binom{a}{t-1} + N'$. Then $\kappa_t(M+N) = \binom{a+1}{t-1} + \kappa_{t-1}(M' + N')$, $\kappa_t M = \binom{a}{t-1} + \kappa_{t-1} M'$, and $\kappa_{t-1} N = \binom{a}{t-2} + \kappa_{t-2} N'$; the result follows by induction on t.

78. [J. Eckhoff and G. Wegner, *Periodica Math. Hung.* **6** (1975), 137–142; A. J. W. Hilton, *Periodica Math. Hung.* **10** (1979), 25–30.] Let $M = |A_1|$ and $N = |A_0|$; we can assume that $t > 0$ and $N > 0$. Then $|\partial A| = |\partial A_1 \cup A_0| + |\partial A_0| \geq \max(|\partial A_1|, |A_0|) + |\partial A_0| \geq \max(\kappa_t M, N) + \kappa_{t-1} N \geq \kappa_t(M + N) = |P_{|A|t}|$, by induction on $m + n + t$.

Conversely, let $A_1 = P_{Mt} + 1$ and $A_0 = P_{N(t-1)} + 1$; this notation means, for example, that $\{210, 320\} + 1 = \{321, 431\}$. Then $\kappa_t(M + N) \le |\partial A| = |\partial A_1 \cup A_0| + |(\partial A_0)0| = \max(\kappa_t M, N) + \kappa_{t-1}N$, because $\partial A_1 = P_{(\kappa_t M)(t-1)} + 1$. [Schützenberger observed in 1959 that $\kappa_t(M+N) \le \kappa_t M + \kappa_{t-1}N$ if and only if $\kappa_t M \ge N$.]

For the first inequality, let A and B be disjoint sets of t-combinations with $|A| = M$, $|\partial A| = \kappa_t M$, $|B| = N$, $|\partial B| = \kappa_t N$. Then $\kappa_t(M+N) = \kappa_t|A \cup B| \le |\partial(A \cup B)| = |\partial A \cup \partial B| = |\partial A| + |\partial B| = \kappa_t M + \kappa_t N$.

79. In fact, $\mu_t(M + \lambda_{t-1}M) = M$, and $\mu_t N + \lambda_{t-1}\mu_t N = N + (n_2 - n_1)[v = 1]$ when N is given by (57).

80. If $N > 0$ and $t > 1$, represent N as in (57) and let $N = N_0 + N_1$, where

$$N_0 = \binom{n_t - 1}{t} + \cdots + \binom{n_v - 1}{v}, \qquad N_1 = \binom{n_t - 1}{t - 1} + \cdots + \binom{n_v - 1}{v - 1}.$$

Let $N_0 = \binom{y}{t}$ and $N_1 = \binom{z}{t-1}$. Then, by induction on t and $\lfloor x \rfloor$, we have $\binom{x}{t} = N_0 + \kappa_t N_0 \ge \binom{y}{t} + \binom{y}{t-1} = \binom{y+1}{t}$; $N_1 = \binom{x}{t} - \binom{y}{t} \ge \binom{x}{t} - \binom{x-1}{t} = \binom{x-1}{t-1}$; and $\kappa_t N = N_1 + \kappa_{t-1}N_1 \ge \binom{z}{t-1} + \binom{z}{t-2} = \binom{z+1}{t-1} \ge \binom{x}{t-1}$.

[Lovász actually proved a stronger result; see exercise 1.2.6–66. We have, similarly, $\mu_t N \ge \binom{x-1}{t-1}$; see Björner, Frankl, and Stanley, *Combinatorica* **7** (1987), 27–28.]

81. For example, if the largest element of $\widehat{P}_{N5}$ is 66433, we have

$$\widehat{P}_{N5} = \{00000, \ldots, 55555\} \cup \{60000, \ldots, 65555\} \cup \{66000, \ldots, 66333\} \cup \{66400, \ldots, 66433\}$$

so $N = \binom{10}{5} + \binom{9}{4} + \binom{6}{3} + \binom{5}{2}$. Its lower shadow is

$$\partial\widehat{P}_{N5} = \{0000, \ldots, 5555\} \cup \{6000, \ldots, 6555\} \cup \{6600, \ldots, 6633\} \cup \{6640, \ldots, 6643\},$$

of size $\binom{9}{4} + \binom{8}{3} + \binom{5}{2} + \binom{4}{1}$.

If the smallest element of Q_{N95} is 66433, we have

$$\widehat{Q}_{N95} = \{99999, \ldots, 70000\} \cup \{66666, \ldots, 66500\} \cup \{66444, \ldots, 66440\} \cup \{66433\}$$

so $N = \bigl(\binom{13}{9}+\binom{12}{8}+\binom{11}{7}\bigr) + \bigl(\binom{8}{6}+\binom{7}{5}\bigr) + \binom{5}{4} + \binom{3}{3}$. Its upper shadow is

$$\varrho\widehat{Q}_{N95} = \{999999, \ldots, 700000\} \cup \{666666, \ldots, 665000\} \\ \cup \{664444, \ldots, 664400\} \cup \{664333, \ldots, 664330\},$$

of size $\bigl(\binom{14}{9}+\binom{13}{8}+\binom{12}{7}\bigr) + \bigl(\binom{9}{6}+\binom{8}{5}\bigr) + \binom{6}{4} + \binom{4}{3} = N + \kappa_9 N$. The size, t, of each combination is essentially irrelevant, as long as $N \le \binom{s+t}{t}$; for example, the smallest element of $\widehat{Q}_{N98}$ is 99966433 in the case we have considered.

82. (a) The derivative would have to be $\sum_{k>0} r_k(x)$, but that series diverges.

[Informally, the graph of $\tau(x)$ shows "pits" of relative magnitude 2^{-k} at all odd multiples of 2^{-k}. Takagi's original publication, in *Proc. Physico-Math. Soc. Japan* (2) **1** (1903), 176–177, has been translated into English in his *Collected Papers* (Iwanami Shoten, 1973).]

(b) Since $r_k(1-t) = (-1)^{\lceil 2^k t \rceil}$ when $k > 0$, we have $\int_0^{1-x} r_k(t)\,dt = \int_x^1 r_k(1-u)\,du = -\int_x^1 r_k(u)\,du = \int_0^x r_k(u)\,du$. The second equation follows from the fact that $r_k(\frac12 t) = r_{k-1}(t)$. Part (d) shows that these two equations suffice to define $\tau(x)$ when x is rational.

(c) Since $\tau(2^{-a}x) = a2^{-a}x + 2^{-a}\tau(x)$ for $0 \le x \le 1$, we have $\tau(\epsilon) = a\epsilon + O(\epsilon)$ when $2^{-a-1} \le \epsilon \le 2^{-a}$. Therefore $\tau(\epsilon) = \epsilon \lg \frac{1}{\epsilon} + O(\epsilon)$ for $0 < \epsilon \le 1$.

(d) Suppose $0 \le p/q \le 1$. If $p/q \le 1/2$ we have $\tau(p/q) = p/q + \tau(2p/q)/2$; otherwise $\tau(p/q) = (q-p)/q + \tau(2(q-p)/q)/2$. Therefore we can assume that q is odd. When q is odd, let $p' = p/2$ when p is even, $p' = (q-p)/2$ when p is odd. Then $\tau(p/q) = 2\tau(p'/q) - 2p'/q$ for $0 < p < q$; this system of $q-1$ equations has a unique solution. For example, the values for $q = 3, 4, 5, 6, 7$ are 2/3, 2/3; 1/2, 1/2, 1/2; 8/15, 2/3, 2/3, 8/15; 1/2, 2/3, 1/2, 2/3, 1/2; 22/49, 30/49, 32/49, 32/49, 30/49, 22/49.

(e) The solutions $< \frac{1}{2}$ are $x = \frac{1}{4}, \frac{1}{4} - \frac{1}{16}, \frac{1}{4} - \frac{1}{16} - \frac{1}{64}, \frac{1}{4} - \frac{1}{16} - \frac{1}{64} - \frac{1}{256}, \ldots, \frac{1}{6}$.

(f) The value $\frac{2}{3}$ is achieved for $x = \frac{1}{2} \pm \frac{1}{8} \pm \frac{1}{32} \pm \frac{1}{128} \pm \cdots$, an uncountable set.

83. Given any integers $q > p > 0$, consider paths starting from 0 in the digraph

$$\begin{array}{ccccccccccccc} 0 & \leftarrow & 1 & \leftarrow & 2 & \leftarrow & 3 & \leftarrow & 4 & \leftarrow & 5 & \leftarrow & \cdots \\ \updownarrow & & \updownarrow & & \updownarrow & & \updownarrow & & \updownarrow & & \updownarrow & & \\ 1 & \rightarrow & 2 & \rightarrow & 3 & \rightarrow & 4 & \rightarrow & 5 & \rightarrow & 6 & \rightarrow & \cdots \end{array}$$

Compute an associated value v, starting with $v \leftarrow -p$; horizontal moves change $v \leftarrow 2v$, vertical moves from node a change $v \leftarrow 2(qa - v)$. The path stops if we reach a node twice with the same value v. Transitions are not allowed to upper node a if $v \le -q$ or $v \ge qa$ at that node; they are not allowed to lower node a with $v \le 0$ or $v \ge q(a+1)$. These restrictions force most steps of the path. (Node a in the upper row means, "Solve $\tau(x) = ax - v/q$"; in the lower row it means, "Solve $\tau(x) = v/q - ax$.") Empirical tests suggest that all such paths are finite. The equation $\tau(x) = p/q$ then has solutions $x = x_0$ defined by the sequence $x_0, x_1, x_2, \ldots$ where $x_k = \frac{1}{2}x_{k+1}$ on a horizontal step and $x_k = 1 - \frac{1}{2}x_{k+1}$ on a vertical step; eventually $x_k = x_j$ for some $j < k$. If $j > 0$ and if q is not a power of 2, these are all the solutions to $\tau(x) = p/q$ when $x > 1/2$.

For example, this procedure establishes that $\tau(x) = 1/5$ and $x > 1/2$ only when x is 83581/87040; the only path yields $x_0 = 1 - \frac{1}{2}x_1$, $x_1 = \frac{1}{2}x_2$, $\ldots$, $x_{18} = \frac{1}{2}x_{19}$, and $x_{19} = x_{11}$. There are, similarly, just two values $x > 1/2$ with $\tau(x) = 3/5$, having denominator $2^{46}(2^{56} - 1)/3$.

Moreover, it appears that all cycles in the digraph that pass through node 0 define values of p and q such that $\tau(x) = p/q$ has uncountably many solutions. Such values are, for example, 2/3, 8/15, 8/21, corresponding to the cycles (01), (0121), (012321). The value 32/63 corresponds to (012121) and also to (012101234545454321), as well as to two other paths that do not return to 0.

84. [Frankl, Matsumoto, Ruzsa, and Tokushige, *J. Combinatorial Theory* **A69** (1995), 125–148.] If $a \le b$ we have

$$\binom{2t-1-b}{t-a}\bigg/T = t^{\underline{a}}(t-1)^{\underline{b-a}}/(2t-1)^{\underline{b}} = 2^{-b}\bigl(1 + f(a,b)t^{-1} + O(b^4/t^2)\bigr),$$

where $f(a,b) = a(1+b) - a^2 - b(1+b)/4 = f(a+1,b) - b + 2a$. Therefore if N has the combinatorial representation (57), and if we set $n_j = 2t - 1 - b_j$, we have

$$\frac{t}{T}\Bigl(\kappa_t N - N\Bigr) = \frac{b_t}{2^{b_t}} + \frac{b_{t-1} - 2}{2^{b_{t-1}}} + \frac{b_{t-2} - 4}{2^{b_{t-2}}} + \cdots + \frac{O(\log t)^3}{t},$$

the terms being negligible when b_j exceeds $2\lg t$. And one can show that

$$\tau\Bigl(\sum_{j=0}^{l} 2^{-e_j}\Bigr) = \sum_{j=0}^{l} (e_j - 2j)2^{-e_j}.$$

85. $N - \lambda_{t-1}N$ has the same asymptotic form as $\kappa_t N - N$, by (63), since $\tau(x) = \tau(1-x)$.

So does $2\mu_t N - N$, up to $O\bigl(T(\log t)^3/t^2\bigr)$, because $\binom{2t-1-b}{t-a} = 2\binom{2t-2-b}{t-a}\bigl(1+O(\log t)/t\bigr)$ when $b < 2\lg t$.

86. $x \in X^{\circ\sim} \iff \bar{x} \notin X^\circ \iff \bar{x} \notin X$ or $\bar{x} \notin X + e_1$ or $\cdots$ or $\bar{x} \notin X + e_n \iff x \in X^\sim$ or $x \in X^\sim - e_1$ or $\cdots$ or $x \in X^\sim - e_n \iff x \in X^{\sim +}$.

87. All three are true, using the fact that $X \subseteq Y^\circ$ if and only if $X^+ \subseteq Y$: (a) $X \subseteq Y^\circ \iff X^\sim \supseteq Y^{\circ\sim} = Y^{\sim +} \iff Y^\sim \subseteq X^{\sim\circ}$. (b) $X^+ \subseteq X^+ \implies X \subseteq X^{+\circ}$; hence $X^\circ \subseteq X^{\circ+\circ}$. Also $X^\circ \subseteq X^\circ \implies X^{\circ+} \subseteq X$; hence $X^{\circ+\circ} \subseteq X^\circ$. (c) $\alpha M \le N \iff S_M^+ \subseteq S_N \iff S_M \subseteq S_N^\circ \iff M \le \beta N$.

88. If $\nu x < \nu y$ then $\nu(x - e_k) < \nu(y - e_j)$, so we can assume that $\nu x = \nu y$ and that $x > y$ in lexicographic order. We must have $y_j > 0$; otherwise $\nu(y - e_j)$ would exceed $\nu(x - e_k)$. If $x_i = y_i$ for $1 \le i \le j$, clearly $k > j$ and $x - e_k \prec y - e_j$. Otherwise $x_i > y_i$ for some $i \le j$; again we have $x - e_k \prec y - e_j$, unless $x - e_k = y - e_j$.

89. From the table

$j =$	0	1	2	3	4	5	6	7	8	9	10	11
$e_j + e_1 =$	e_1	e_0	e_4	e_5	e_2	e_3	e_8	e_9	e_6	e_7	e_{11}	e_{10}
$e_j + e_2 =$	e_2	e_4	e_0	e_6	e_1	e_8	e_3	e_{10}	e_5	e_{11}	e_7	e_9
$e_j + e_3 =$	e_3	e_5	e_6	e_7	e_8	e_9	e_{10}	e_0	e_{11}	e_1	e_2	e_4

we find $(\alpha 0, \alpha 1, \ldots, \alpha 12) = (0,4,6,7,8,9,10,11,11,12,12,12,12)$; $(\beta 0, \beta 1, \ldots, \beta 12) = (0,0,0,0,1,1,2,3,4,5,6,8,12)$.

90. Let $Y = X^+$ and $Z = C_k X$, and let $N_a = |X_k(a)|$ for $0 \le a < m_k$. Then

$$|Y| = \sum_{a=0}^{m_k-1} |Y_k(a)| = \sum_{a=0}^{m_k-1} \bigl|(X_k(a-1)+e_k) \cup (X_k(a)+E_k(0))\bigr|$$
$$\ge \sum_{a=0}^{m_k-1} \max(N_{a-1}, \alpha N_a),$$

where $a-1$ stands for $(a-1) \bmod m_k$ and the α function comes from the $(n-1)$-dimensional torus, because $|X_k(a) + E_k(0)| \ge \alpha N_a$ by induction. Also

$$|Z^+| = \sum_{a=0}^{m_k-1} |Z_k^+(a)| = \sum_{a=0}^{m_k-1} \bigl|(Z_k(a-1)+e_k) \cup (Z_k(a)+E_k(0))\bigr|$$
$$= \sum_{a=0}^{m_k-1} \max(N_{a-1}, \alpha N_a),$$

because both $Z_k(a-1)+e_k$ and $Z_k(a)+E_k(0)$ are standard in $n-1$ dimensions.

91. Let there be N_a points in row a of a totally compressed array, where row 0 is at the bottom; thus $l = N_{-1} \ge N_0 \ge \cdots \ge N_{m-1} \ge N_m = 0$. We show first that there is an optimum X for which the "bad" condition $N_a = N_{a+1}$ never occurs except when $N_a = 0$ or $N_a = l$. For if a is the smallest bad subscript, suppose $N_{a-1} > N_a = N_{a+1} = \cdots = N_{a+k} > N_{a+k+1}$. Then we can always decrease N_{a+k} by 1 and add 1 to some N_b for $b \le a$ without increasing $|X^+|$, except in cases where $k = 1$ and $N_{a+2} = N_{a+1} - 1$ and $N_b = N_a + a - b < l$ for $0 \le b \le a$. Exploring such cases further, if $N_{c+1} < N_c = N_{c-1}$ for some $c > a+1$, we can set $N_c \leftarrow N_c - 1$ and $N_a \leftarrow N_a + 1$, thereby either decreasing a or increasing N_0. Otherwise we can find a subscript d such that $N_c = N_{a+1} + a + 1 - c > 0$ for $a < c < d$, and either $N_d = 0$ or $N_d < N_{d-1} - 1$. Then it is OK to decrease N_c by 1 for $a < c < d$ and subsequently to increase N_b by 1 for $0 \le b < d-a-1$. (It is important to note that if $N_d = 0$ we have $N_0 \ge d-1$; hence $d = m$ implies $l = m$.)

Repeating such transformations until $N_a > N_{a+1}$ whenever $N_a \ne l$ and $N_{a+1} \ne 0$, we reach situation (86), and the proof can be completed as in the text.

92. Let $x+k$ denote the lexicographically smallest element of $T(m_1, \ldots, m_{n-1})$ that exceeds x and has weight $\nu x + k$, if any such element exists. For example, if $m_1 = m_2 = m_3 = 4$ and $x = 211$, we have $x+1 = 212$, $x+2 = 213$, $x+3 = 223$, $x+4 = 233$, $x+5 = 333$, and $x+6$ does not exist; in general, $x+k+1$ is obtained from $x+k$ by increasing the rightmost component that can be increased. If $x+k = (m_1-1, \ldots, m_{n-1}-1)$, let us set $x+k+1 = x+k$. Then if $S(k)$ is the set of all elements of $T(m_1, \ldots, m_{n-1})$ that are $\preceq x+k$, we have $S(k+1) = S(k)^+$. Furthermore, the elements of S that end in a are those whose first $n-1$ components are in $S(m-1-a)$.

The result of this exercise can be stated more intuitively: As we generate n-dimensional standard sets S_1, S_2, $\ldots$, the $(n-1)$-dimensional standard sets on each layer become spreads of each other just after each point is added to layer $m-1$. Similarly, they become cores of each other just before each point is added to layer 0.

93. (a) Suppose the parameters are $2 \le m'_1 \le m'_2 \le \cdots \le m'_n$ when sorted properly, and let k be minimal with $m_k \ne m'_k$. Then take $N = 1 + \mathrm{rank}(0, \ldots, 0, m'_k - 1, 0, \ldots, 0)$. (We must assume that $\min(m_1, \ldots, m_n) \ge 2$, since parameters equal to 1 can be placed anywhere.)

(b) Only in the proof for $n = 2$, buried inside the answer to exercise 91. That proof is incorporated by induction when n is larger.

94. Complementation reverses lexicographic order and changes ϱ to ∂.

95. For Theorem K, let $d = n-1$ and $s_0 = \cdots = s_d = 1$. For Theorem M, let $d = s$ and $s_0 = \cdots = s_d = t+1$.

96. In such a representation, N is the number of t-multicombinations of $\{s_0 \cdot 0, s_1 \cdot 1, s_2 \cdot 2, \ldots\}$ that precede $n_t n_{t-1} \ldots n_1$ in lexicographic order, because the generalized coefficient $\binom{S(n)}{t}$ counts the multicombinations whose leftmost component is $< n$.

If we truncate the representation by stopping at the rightmost nonzero term $\binom{S(n_v)}{v}$, we obtain a nice generalization of (60):

$$|\partial P_{Nt}| = \binom{S(n_t)}{t-1} + \binom{S(n_{t-1})}{t-2} + \cdots + \binom{S(n_v)}{v-1}.$$

[See G. F. Clements, *J. Combinatorial Theory* **A37** (1984), 91–97. The inequalities $s_0 \ge s_1 \ge \cdots \ge s_d$ are needed for the validity of Corollary C, but not for the calculation of $|\partial P_{Nt}|$. Some terms $\binom{S(n_k)}{k}$ for $t \ge k > v$ may be zero. For example, when $N = 1$, $t = 4$, $s_0 = 3$, and $s_1 = 2$, we have $N = \binom{S(1)}{4} + \binom{S(1)}{3} = 0 + 1$.]

97. (a) The tetrahedron has four vertices, six edges, four faces: $(N_0, \ldots, N_4) = (1, 4, 6, 4, 1)$. The octahedron, similarly, has $(N_0, \ldots, N_6) = (1, 6, 8, 8, 0, 0, 0)$, and the icosahedron has $(N_0, \ldots, N_{12}) = (1, 12, 30, 20, 0, \ldots, 0)$. The hexahedron, aka the 3-cube, has eight vertices, 12 edges, and six square faces; perturbation breaks each square face into two triangles and introduces new edges, so we have $(N_0, \ldots, N_8) = (1, 8, 18, 12, 0, \ldots, 0)$. Finally, the perturbed pentagonal faces of the dodecahedron lead to $(N_0, \ldots, N_{20}) = (1, 20, 54, 36, 0, \ldots, 0)$.

(b) $\{210, 310\} \cup \{10, 20, 21, 30, 31\} \cup \{0, 1, 2, 3\} \cup \{\epsilon\}$.

(c) $0 \le N_t \le \binom{n}{t}$ for $0 \le t \le n$ and $N_{t-1} \ge \kappa_t N_t$ for $1 \le t \le n$. The second condition is equivalent to $\lambda_{t-1} N_{t-1} \ge N_t$ for $1 \le t \le n$, if we define $\lambda_0 1 = \infty$. These conditions are necessary for Theorem K, and sufficient if $A = \bigcup P_{N_t t}$.

(d) The complements of the elements not in a simplicial complex, namely the sets $\{\{0,\ldots,n-1\}\setminus\alpha \mid \alpha\notin C\}$, form a simplicial complex. (We can also verify that the necessary and sufficient condition holds: $N_{t-1}\ge\kappa_t N_t \iff \lambda_{t-1}N_{t-1}\ge N_t \iff \kappa_{n-t+1}\overline{N}_{n-t+1}\le\overline{N}_{n-t}$, because $\kappa_{n-t}\overline{N}_{n-t+1}=\binom{n}{t}-\lambda_{t-1}N_{t-1}$ by exercise 94.)

(e) 00000 $\leftrightarrow$ 14641; 10000 $\leftrightarrow$ 14640; 11000 $\leftrightarrow$ 14630; 12000 $\leftrightarrow$ 14620; 13000 $\leftrightarrow$ 14610; 14000 $\leftrightarrow$ 14600; 12100 $\leftrightarrow$ 14520; 13100 $\leftrightarrow$ 14510; 14100 $\leftrightarrow$ 14500; 13200 $\leftrightarrow$ 14410; 14200 $\leftrightarrow$ 14400; 13300 $\leftrightarrow$ 14400; and the self-dual cases 14300, 13310.

98. The following procedure by S. Linusson [*Combinatorica* **19** (1999), 255–266], who considered also the more general problem for multisets, is considerably faster than a more obvious approach. Let $L(n,h,l)$ count feasible vectors with $N_t=\binom{n}{t}$ for $0\le t\le l$, $N_{t+1}<\binom{n}{t+1}$, and $N_t=0$ for $t>h$. Then $L(n,h,l)=0$ unless $-1\le l\le h\le n$; also $L(n,h,h)=L(n,h,-1)=1$, and $L(n,n,l)=L(n,n-1,l)$ for $l<n$. When $n>h\ge l\ge 0$ we can compute $L(n,h,l)=\sum_{j=l}^{h}L(n-1,h,j)L(n-1,j-1,l-1)$, a recurrence that follows from Theorem K. (Each size vector corresponds to the complex $\bigcup P_{N_t t}$, with $L(n-1,h,j)$ representing combinations that do not contain the maximum element $n-1$ and $L(n-1,j-1,l-1)$ representing those that do.) Finally the grand total is $L(n)=\sum_{l=1}^{n}L(n,n,l)$.

We have $L(0)$, $L(1)$, $L(2)$, $\ldots$ = 2, 3, 5, 10, 26, 96, 553, 5461, 100709, 3718354, 289725509, $\ldots$; $L(100)\approx 3.2299\times10^{1842}$.

99. The maximal elements of a simplicial complex form a clutter; conversely, the combinations contained in elements of a clutter form a simplicial complex. Thus the two concepts are essentially equivalent.

(a) If $(M_0,M_1,\ldots,M_n)$ is the size vector of a clutter, then $(N_0,N_1,\ldots,N_n)$ is the size vector of a simplicial complex if $N_n=M_n$ and $N_t=M_t+\kappa_{t+1}N_{t+1}$ for $0\le t<n$. Conversely, every such $(N_0,\ldots,N_n)$ yields an $(M_0,\ldots,M_n)$ if we use the lexicographically first N_t t-combinations. [G. F. Clements extended this result to general multisets in *Discrete Math.* **4** (1973), 123–128.]

(b) In the order of answer 97(e) they are 00000, 00001, 10000, 00040, 01000, 00030, 02000, 00120, 03000, 00310, 04000, 00600, 00100, 00020, 01100, 00210, 02100, 00500, 00200, 00110, 01200, 00400, 00300, 01010, 01300, 00010. Notice that $(M_0,\ldots,M_n)$ is feasible if and only if $(M_n,\ldots,M_0)$ is feasible, so we have a different sort of duality in this interpretation.

100. Represent A as a subset of $T(m_1,\ldots,m_n)$ as in the proof of Corollary C. Then the maximum value of νA is obtained when A consists of the N lexicographically smallest points $x_1\ldots x_n$.

The proof starts by reducing to the case that A is compressed, in the sense that its t-multicombinations are $P_{|A\cap T_t|t}$ for each t. Then if y is the largest element $\in A$ and if x is the smallest element $\notin A$, we prove that $x<y$ implies $\nu x>\nu y$, hence $\nu(A\setminus\{y\}\cup\{x\})>\nu A$. For if $\nu x=\nu y-k$ we could find an element of $\partial^k y$ that is greater than x, contradicting the assumption that A is compressed.

101. (a) In general, $F(p)=N_0p^n+N_1p^{n-1}(1-p)+\cdots+N_n(1-p)^n$ when $f(x_1,\ldots,x_n)$ is satisfied by exactly N_t binary strings $x_1\ldots x_n$ of weight t. Thus we find $G(p)=p^4+3p^3(1-p)+p^2(1-p)^2$; $H(p)=p^4+p^3(1-p)+p^2(1-p)^2$.

(b) A monotone formula f is equivalent to a simplicial complex C under the correspondence $f(x_1,\ldots,x_n)=1\iff\{j-1\mid x_j=0\}\in C$. Therefore the functions $f(p)$ of monotone Boolean functions are those that satisfy the condition of exercise 97(c), and we obtain a suitable function by choosing the lexicographically last N_{n-t} t-combinations (which are complements of the first N_s s-combinations): $\{3210\}$, $\{321, 320, 310\}$, $\{32\}$ gives $f(w,x,y,z) = wxyz \vee xyz \vee wyz \vee wxz \vee yz = wxz \vee yz$.

M. P. Schützenberger observed that we can find the parameters N_t easily from $f(p)$ by noting that $f\bigl(1/(1+u)\bigr) = (N_0 + N_1u + \cdots + N_nu^n)/(1+u)^n$. One can show that $H(p)$ is not equivalent to a monotone formula in any number of variables, because $(1+u+u^2)/(1+u)^4 = (N_0 + N_1u + \cdots + N_nu^n)/(1+u)^n$ implies that $N_1 = n-3$, $N_2 = \binom{n-3}{2} + 1$, and $\kappa_2 N_2 = n-2$.

But the task of deciding this question is not so simple in general. For example, the function $(1 + 5u + 5u^2 + 5u^3)/(1+u)^5$ does not match any monotone formula in five variables, because $\kappa_3 5 = 7$; but it equals $(1 + 6u + 10u^2 + 10u^3 + 5u^4)/(1+u)^6$, which works fine with six.

102. (a) Choose N_t linearly independent polynomials of degree t in I; order their terms lexicographically, and take linear combinations so that the lexicographically smallest terms are distinct monomials. Let I' consist of all multiples of those monomials.

(b) Each monomial of degree t in I' is essentially a t-multicombination; for example, $x_1^3x_2x_5^4$ corresponds to 55552111. If M_t is the set of independent monomials for degree t, the ideal property is equivalent to saying that $M_{t+1} \supseteq \varrho M_t$.

In the given example, $M_3 = \{x_0x_1^2\}$; $M_4 = \varrho M_3 \cup \{x_0x_1x_2^2\}$; $M_5 = \varrho M_4 \cup \{x_1x_2^4\}$, since $x_2^2(x_0x_1^2 - 2x_1x_2^2) - x_1(x_0x_1x_2^2) = -2x_1x_2^4$; and $M_{t+1} = \varrho M_t$ thereafter.

(c) By Theorem M we can assume that $M_t = \widehat{Q}_{Mst}$. Let $N_t = \binom{n_{ts}}{s} + \cdots + \binom{n_{t2}}{2} + \binom{n_{t1}}{1}$, where $s + t \geq n_{ts} > \cdots > n_{t2} > n_{t1} \geq 0$; then $n_{ts} = s + t$ if and only if $n_{t(s-1)} = s-2, \ldots, n_{t1} = 0$. Furthermore we have

$$N_{t+1} \geq N_t + \kappa_s N_t = \binom{n_{ts} + [n_{ts} \geq s]}{s} + \cdots + \binom{n_{t2} + [n_{t2} \geq 2]}{2} + \binom{n_{t1} + [n_{t1} \geq 1]}{1}.$$

Therefore the sequence $\bigl(n_{ts} - t - \infty[n_{ts} < s], \ldots, n_{t2} - t - \infty[n_{t2} < 2], n_{t1} - t - \infty[n_{t1} < 1]\bigr)$ is lexicographically nondecreasing as t increases, where we insert '$-\infty$' in components that have $n_{tj} = j-1$. Such a sequence cannot increase infinitely many times without exceeding the maximum value $(s, -\infty, \ldots, -\infty)$, by exercise 1.2.1–15(d).

103. Let P_{Nst} be the first N elements of a sequence determined as follows: For each binary string $x = x_{s+t-1} \ldots x_0$, in lexicographic order, write down $\binom{\nu x}{t}$ subcubes by changing t of the 1s to $*$s in all possible ways, in lexicographic order (considering $1 < *$). For example, if $x = 0101101$ and $t = 2$, we generate the subcubes 0101∗0∗, 010∗10∗, 010∗∗01, 0∗0110∗, 0∗01∗01, 0∗0∗101.

[See B. Lindström, *Arkiv för Mat.* **8** (1971), 245–257; a generalization analogous to Corollary C appears in K. Engel, *Sperner Theory* (Cambridge Univ. Press, 1997), Theorem 8.1.1.]

104. The first N strings in cross order have the desired property. [T. N. Danh and D. E. Daykin, *J. London Math. Soc.* (2) **55** (1997), 417–426.]

Notes: Beginning with the observation that the "1-shadow" of the N lexicographically first strings of weight t (namely the strings obtained by deleting 1 bits only) consists of the first $\mu_t N$ strings of weight t, R. Ahlswede and N. Cai extended the Danh–Daykin theorem to allow insertion, deletion, and/or transposition of bits [*Combinatorica* **17** (1997), 11–29; *Applied Math. Letters* **11**, 5 (1998), 121–126]. Uwe Leck has proved that no total ordering of *ternary strings* has the analogous minimum-shadow property [Preprint 98/6 (Univ. Rostock, 1998), 6 pages].

105. Every number must occur the same number of times in the cycle. Equivalently, $\binom{n-1}{t-1}$ must be a multiple of t. This necessary condition appears to be sufficient as well, provided that n is not too small with respect to t; but such a result may well be true yet impossible to prove. [See Chung, Graham, and Diaconis, *Discrete Math.* **110** (1992), 55–57.]

The next few exercises consider the cases $t = 2$ and $t = 3$, for which elegant results are known. Similar but more complicated results have been derived for $t = 4$ and $t = 5$, and the case $t = 6$ has been partially resolved. The case $(n, t) = (12, 6)$ is currently the smallest for which the existence of a universal cycle is unknown.

106. Let the differences mod $(2m+1)$ be $1, 2, \ldots, m, 1, 2, \ldots, m, \ldots$, repeated $2m+1$ times; for example, the cycle for $m = 3$ is (013602561450346235124). This works because $1 + \cdots + m = \binom{m+1}{2}$ is relatively prime to $2m + 1$. [*J. École Polytechnique* **4**, Cahier 10 (1810), 16–48.]

107. The seven doubles , , ...,  can be inserted in 3^7 ways into any universal cycle of 3-combinations for $\{0, 1, 2, 3, 4, 5, 6\}$. The number of such universal cycles is the number of Eulerian trails of the complete graph K_7, which can be shown to be 129,976,320 if we regard $(a_0a_1 \ldots a_{20})$ as equivalent to $(a_1 \ldots a_{20}a_0)$ but not to the reverse-order cycle $(a_{20} \ldots a_1a_0)$. So the answer is 284,258,211,840.

[This problem was first solved in 1859 by M. Reiss, whose method was so complicated that people doubted the result; see *Nouvelles Annales de Mathématiques* **8** (1849), 74; **11** (1852), 115; *Annali di Matematica Pura ed Applicata* (2) **5** (1871–1873), 63–120. A considerably simpler solution, confirming Reiss's claim, was found by P. Jolivald and G. Tarry, who also enumerated the Eulerian trails of K_9; see *Comptes Rendus Association Française pour l'Avancement des Sciences* **15**, part 2 (1886), 49–53; É. Lucas, *Récréations Mathématiques* **4** (1894), 123–151. Brendan D. McKay and Robert W. Robinson found an approach that is better still, enabling them to continue the enumeration through K_{21} by using the fact that the number of trails is

$$(m-1)!^{2m+1}\,[z_0^{2m} z_1^{2m-2} \ldots z_{2m}^{2m-2}]\ \det(a_{jk}) \prod_{1 \le j < k \le 2m} (z_j^2 + z_k^2),$$

where $a_{jk} = -1/(z_j^2 + z_k^2)$ when $j \ne k$; $a_{jj} = -1/(2z_j^2) + \sum_{0 \le k \le 2m} 1/(z_j^2 + z_k^2)$; see *Combinatorics, Probability, and Computing* **7** (1998), 437–449.]

C. Flye Sainte-Marie, in L'*Intermédiaire des Mathématiciens* **1** (1894), 164–165, noted that the Eulerian trails of K_7 include 2×720 that have 7-fold symmetry under permutation of $\{0, 1, \ldots, 6\}$ (namely Poinsot's cycle and its reverse), plus 32×1680 with 3-fold symmetry, plus 25778×5040 cycles that are asymmetric.

108. No solution is possible for $n < 7$, except in the trivial case $n = 4$. When $n = 7$ there are $12{,}255{,}208 \times 7!$ universal cycles, not considering $(a_0a_1 \ldots a_{34})$ to be the same as $(a_1 \ldots a_{34}a_0)$, including cases with 5-fold symmetry like the example cycle in exercise 105.

When $n \ge 8$ we can proceed systematically as suggested by B. Jackson in *Discrete Math.* **117** (1993), 141–150; see also G. Hurlbert, *SIAM J. Disc. Math.* **7** (1994), 598–604: Put each 3-combination into the "standard cyclic order" $c_1c_2c_3$ where $c_2 = (c_1 + \delta) \bmod n$, $c_3 = (c_2 + \delta') \bmod n$, $0 < \delta, \delta' < n/2$, and either $\delta = \delta'$ or $\max(\delta, \delta') < n - \delta - \delta' \ne (n-1)/2$ or $(1 < \delta < n/4$ and $\delta' = (n-1)/2)$ or $(\delta = (n-1)/2$ and $1 < \delta' < n/4)$. For example, when $n = 8$ the allowable values of (δ, δ') are $(1,1)$, $(1,2)$, $(1,3)$, $(2,1)$, $(2,2)$, $(3,1)$, $(3,3)$; when $n = 11$ they are $(1,1)$, $(1,2)$, $(1,3)$, $(1,4)$, $(2,1)$, $(2,2)$, $(2,3)$, $(2,5)$, $(3,1)$, $(3,2)$, $(3,3)$, $(4,1)$, $(4,4)$, $(5,2)$, $(5,5)$. Then construct the digraph with vertices (c, δ) for $0 \le c < n$ and $1 \le \delta < n/2$, and with arcs $(c_1, \delta) \to (c_2, \delta')$ for every combination $c_1c_2c_3$ in standard cyclic order. This digraph is connected and balanced, so it has an Eulerian trail by Theorem 2.3.4.2D. (The peculiar rules about $(n-1)/2$ make the digraph connected when n is odd. The Eulerian trail can be chosen to have n-fold symmetry when $n=8$, but not when $n=12$.)

109. When $n=1$ the cycle (000) is trivial; when $n=2$ there is no cycle; and there are essentially only two when $n=4$, namely (00011122233302021313) and (00011120203332221313). When $n \geq 5$, let the multicombination $d_1d_2d_3$ be in standard cyclic order if $d_2 = (d_1+\delta-1) \bmod n$, $d_3 = (d_2+\delta'-1) \bmod n$, and (δ,δ') is allowable for $n+3$ in the previous answer. Construct the digraph with vertices (d,δ) for $0 \leq d < n$ and $1 \leq \delta < (n+3)/2$, and with arcs $(d_1,\delta) \to (d_2,\delta')$ for every multicombination $d_1d_2d_3$ in standard cyclic order; then find an Eulerian trail.

Perhaps a universal cycle of t-multicombinations exists for $\{0,1,\ldots,n-1\}$ if and only if a universal cycle of t-combinations exists for $\{0,1,\ldots,n+t-1\}$.

110. A nice way to check for runs is to compute the numbers $b(S) = \sum\{2^{p(c)} \mid c \in S\}$ where $\bigl(p(\mathtt{A}),\ldots,p(\mathtt{K})\bigr) = (1,\ldots,13)$; then set $l \leftarrow b(S) \wedge -b(S)$ and check that $b(S)+l = l \ll s$, and also that $\bigl((l \ll s) \vee (l \gg 1)\bigr) \wedge a = 0$, where $a = 2^{p(c_1)} \vee \cdots \vee 2^{p(c_5)}$. The values of $b(S)$ and $\sum\{v(c) \mid c \in S\}$ are easily maintained as S runs through all 31 nonempty subsets in Gray-code order. The answers are (1009008, 99792, 2813796, 505008, 2855676, 697508, 1800268, 751324, 1137236, 361224, 388740, 51680, 317340, 19656, 90100, 9168, 58248, 11196, 2708, 0, 8068, 2496, 444, 356, 3680, 0, 0, 0, 76, 4) for $x = (0,\ldots,29)$; thus the mean score is ≈ 4.769 and the variance is ≈ 9.768.

Hands without points are sometimes facetiously called nineteen,
as that number cannot be made by the cards.
— G. H. DAVIDSON, *Dee's Hand-Book of Cribbage* (1839)

Note: A four-card flush is not allowed in the "crib." Then the distribution is a bit easier to compute, and it turns out to be (1022208, 99792, 2839800, 508908, 2868960, 703496, 1787176, 755320, 1118336, 358368, 378240, 43880, 310956, 16548, 88132, 9072, 57288, 11196, 2264, 0, 7828, 2472, 444, 356, 3680, 0, 0, 0, 76, 4); the mean and variance decrease to approximately 4.735 and 9.667.

INDEX AND GLOSSARY

When an index entry refers to a page containing a relevant exercise, see also the *answer* to that exercise for further information. An answer page is not indexed here unless it refers to a topic not included in the statement of the exercise.

2-nomial coefficients, 38.
κ_t (Kruskal function), 19–21, 31–34, 51.
λ_t (Kruskal function), 20–21, 32–33.
μ_t (Macaulay function), 20–21, 32–33, 51.
ν (sideways sum), 20, 29, 37.
π (circle ratio), as "random" example, 2, 13, 27–29, 35.
τ (Takagi function), 20–21, 32–33.
∂ (shadow), 18.
ϱ (upper shadow), 18.

Active bits, 12.
Adjacent transpositions, 15–17, 30.
Ahlswede, Rudolph, 56.
Alternating combinatorial number system, 9, 27.
Analysis of algorithms, 4–5, 25, 27, 29.
Antichains of subsets, *see* Clutters.

Balanced ternary notation, 41.
Baseball, 26.
Basis of vector space, 26, 31.
Basis theorem, 34.
Beckenbach, Edwin Ford, 5.
Bellman, Richard Ernest, 19.
Bernoulli, Jacques (= Jakob = James), iii, 16.
Binary tree representation of tree, 27.
Binary vector spaces, 26, 31.
Binomial coefficients, 1, 32.
  generalized, 33.
Binomial number system, *see* Combinatorial number system.
Binomial trees, 6–7, 27.
Bitner, James Richard, 8.
Bitwise manipulation, 4, 58.
Björner, Anders, 51.
Boolean functions, 34.
Bounded compositions, 16, 30, 31.
Buck, Marshall Wilbert, 30.

Cai, Ning (蔡宁), 56.
Calabi, Eugenio, 38.
Canonical bases, 26, 31.
Caron, Jacques, 42.
Catalan, Eugène Charles, 36.
Chase, Phillip John, 11–13, 16, 28–29, 45.
Chinese rings, 28.
Chords, 10, 30.
Chung Graham, Fan Rong King (鍾金芳蓉), 57.
Clements, George Francis, 24–25, 34, 54, 55.
Cliques, 31.
Clutters, 34.
Colex order, 5.
Combination generation, 1–18, 25–31, 35.
  Gray codes for, 8–18.
  homogeneous, 10–11, 16–17, 28–29, 41, 45, 48.
  near-perfect, 11–17, 29.
  perfect, 15–17, 30.
Combinations, 1–35.
  dual, 2–4, 26–27, 29.
  of a multiset, 2–3, 16–18, 25, 33.
  with repetitions, 2–3, 11, 16–19, 25, 33.
Combinatorial number system, 6, 27, 31–32, 37.
  alternating, 9, 27.
  generalized, 33.
Complement in a torus, 21.
Complete binary tree, 39.
Complete graph, 57.
Compositions, 2–4, 11, 25, 38.
  bounded, 16, 30, 31.
Compression of a set, 23, 33, 55.
Contingency tables, 18, 31.
Core set in a torus, 22–23, 33.
Cribbage, 35.
Cross-intersecting sets, 31.
Cross order, 20–25, 33, 56.
Cycle, universal, of combinations, 35.
Czerny, Carl, 47.

Danh, Tran-Ngoc, 56.
Davidson, George Henry, 58.
Daykin, David Edward, 50, 56.
De Morgan, Augustus, 1.
Delta sequences, 46, 47.
Derivative, 32.
Diaconis, Persi Warren, 57.
Dimension of a vector space, 26.
Dominoes, 35.
Dual combinations, 2–4, 26–27, 29.
Dual set in a torus, 22–23.
Dual size vector, 34.
Duality, 33, 55.
Dvořák, Stanislav, 37.

Eades, Peter Dennis, 16, 46.
Eckhoff, Jürgen, 50.
Ehrlich, Gideon (גדעון ארליך), 8, 42.
End-around swaps, 30.
Endo-order, 14, 29.
Engel, Konrad Wolfgang, 56.
Enns, Theodore Christian, 46.

59

Erdős, Pál (= Paul), 19.
Euler, Leonhard (Ейлеръ, Леонардъ = Эйлер, Леонард), trails, 57, 58.

Fenichel, Robert Ross, 25.
First-element swaps, 16–17, 30.
Flye Sainte-Marie, Camille, 57.
Fraenkel, Aviezri S (אביעזרי פרנקל), 39.
Frankl, Péter, 51, 52.

Generating functions, 29, 47.
Genlex order, 9–13, 16–17, 28–29, 44, 48.
  for Gray codes, 31.
Golomb, Solomon Wolf, 2, 25.
Graham, Ronald Lewis (葛立恒), 57.
Gray, Frank, binary code, 8, 49, 58.
  codes for combinations, 8–18, 27–30.
Grid paths, 2–3, 25.

Hamilton, William Rowan, cycles, 8, 46.
  paths, 16, 30, 46, 48.
Hickey, Thomas Butler, 16, 46.
Hilbert, David, basis theorem, 34.
Hilton, Anthony John William, 31, 50.
Homogeneous generation, 10–11, 28–30, 45.
  scheme K_{st}, 10, 16–17, 29, 41, 48.
Homogeneous polynomials, 34.
Hurlbert, Glenn Howland, 57.
Hypergraphs, 18.

Internet, ii, iii, 26.
Ising, Ernst, configurations, 26, 31, 38.
Iteration versus recursion, 12–14, 29.

Jackson, Bradley Warren, 57.
Jenkyns, Thomas Arnold, 11.
Jolivald, Philippe (= Paul de Hijo), 57.

Katona, Gyula (Optimális Halmaz), 19.
Keyboard, 10, 30.
Knapsack problem, 7.
Knuth, Donald Ervin (高德纳), i, iv, 38.
Korsh, James F., 38.
Kruskal, Joseph Bernard, Jr., 19–20.
  function κ_t, 19–21, 31–34, 51.
  function λ_t, 20–21, 32–33.
  –Katona theorem, 19.

Lattice paths, 2–3, 25.
Leck, Uwe, 56.
Lehmer, Derrick Henry, 5, 30, 46.
Lexicographic generation, 4–7, 16–19, 25–27, 29, 31, 47.
Lindström, Bernt Lennart Daniel, 24–25, 34, 56.
Linked lists, 27, 39.
Linusson, Hans Svante, 55.
Lipschutz, Seymour Saul, 38.
Liu, Chao-Ning (劉兆寧), 8.

Loopless generation, 8, 25, 27, 28, 41, 45, 46.
Lovász, László, 32, 51.
Lucas, François Édouard Anatole, 57.
Lüneburg, Heinz, 39.

Macaulay, Francis Sowerby, 19, 34, 50.
  function μ_t, 20–21, 32–33, 51.
Matrix multiplication, 43.
Matsumoto, Makoto (松本眞), 52.
McCarthy, David, 11.
McKay, Brendan Damien, 57.
Middle levels conjecture, 47.
Min-plus matrix multiplication, 43.
`MMIX`, ii.
Monomials, 34.
Monotone Boolean functions, 34.
Mor, Moshe (משה מור), 39.
Multicombinations: Combinations with repetitions, 2–3, 11, 16–19, 25, 33.
Multisets, 2, 36.
  combinations of, 2–3, 16–18, 25, 33.
  permutations of, 4, 14–15, 29, 30, 38.

Near-perfect combination generation, 11–17, 29.
Near-perfect permutation generation, 15, 29.
Nijenhuis, Albert, 8.
Nowhere differentiable function, 32.

Olive, Gloria, 46.
Order ideal, 33.
Organ-pipe order, 14.

Partitions, 38.
  ordered, *see* Compositions.
Pascal, Ernesto, 6.
Paths on a grid, 2–3, 25.
Payne, William Harris, 9, 28.
Perfect combination generation, 15–17, 30.
Permutations of multisets, 4, 14–15, 29, 30, 38.
Pi (π), as "random" example, 2, 13, 27–29, 35.
Piano, 10, 30.
Plain changes, 10.
Playing cards, 35.
Poinsot, Louis, 35, 57.
Polyhedron, 18, 33.
Polynomial ideal, 34.
Postorder traversal, 27.
Preorder traversal, 7, 27, 43.

q-multinomial coefficients, 30.
q-nomial coefficients, 15, 30, 38, 47.

Rademacher, Hans, functions, 32.
Ranking a combination, 6, 9, 19, 29, 39, 40.
Read, Ronald Cedric, 16, 46.
Recurrences, 26, 40–42.
Recursion, 10–12.
  versus iteration, 12–14, 29.
Recursive coroutines, 16.
Reflected Gray code, 28.
Regular solids, 33.
Reingold, Edward Martin (ריינגולד,
  יצחק משה בן חיים), 8.
Reiss, Michel, 57.
Replacement selection sorting, 39.
Reversion of power series, 39.
Revolving door property, 8, 29–30.
  scheme Γ_{st}, 8–10, 16–17, 27–29.
Robinson, Robert William, 57.
Root of unity, 30.
Row-echelon form, 37.
Rucksack filling, 7, 27.
Ruskey, Frank, 30.
Ruzsa, Imre Zoltán, 52.

Savage, Carla Diane, 47.
Schützenberger, Marcel Paul, 19, 51, 56.
Shadows, 18–25, 31–34.
  of binary strings, 35.
  of subcubes, 34.
Shields, Ian Beaumont, 47.
Sibling links, 27.
Sideways sum, 20, 29, 37.
Simões Pereira, José Manuel dos Santos, 38.
Simplexes, 18.
Simplicial complexes, 33–34, 55.
Simplicial multicomplexes, 34.
Size vectors, 33, 34.
Sperner, Emanuel, theory, 56.
Spread set in a torus, 22–25, 33.
Stachowiak, Grzegorz, 46.
Standard set in a torus, 22–24, 33.
Stanford GraphBase, ii, iii.
Stanley, Richard Peter, 14, 51.
Star transpositions, 16–17, 30.
Subcubes, 31, 34.
Swapping with the first element, 16–17, 30.

Takagi, Teiji (高木貞治), 20, 51.
  function, 20–21, 32–33.
Tang, Donald Tao-Nan (唐道南), 8.
Tarry, Gaston, 57.
Ternary strings, 28, 56.
Terquem, Olry, 35.
Tokushige, Norihide (徳重典英), 52.
Topological sorting, 46.
Török, Éva, 45.
Torus, n-dimensional, 20–25, 33.
Tree of losers, 39.
Triangles, 20.
Triangulation, 37.
Trie, 9–10.

Unit vectors, 22.
Universal cycles of combinations, 35.
Unranking a combination, 27, 29.
Upper shadow, 18.

van Zanten, Arend Jan, 40.
Vector spaces, 26, 31.

Walsh, Timothy Robert Stephen, 9, 48.
Wang, Da-Lun (王大倫), 20, 22.
Wang, Ping Yang (王平, née 楊平), 20, 22.
Wegner, Gerd, 50.
Whipple, Francis John Welsh, 22.
Wiedemann, Douglas Henry, 30.
Wilf, Herbert Saul, 8, 38.
Williams, Aaron Michael, 46.

z-nomial coefficients, 15, 30, 38, 47.
Zanten, Arend Jan van, 40.