Note to readers:
Please ignore these sidenotes; they're just hints to myself for preparing the index, and they're often flaky!

KNUTH

THE ART OF COMPUTER PROGRAMMING

VOLUME 4 PRE-FASCICLE 5C

DANCING LINKS

DONALD E. KNUTH *Stanford University*

ADDISON–WESLEY

August 8, 2017

Internet
Stanford GraphBase
MMIX

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.

See also http://www-cs-faculty.stanford.edu/~knuth/programs.html for various experimental programs that I wrote while writing this material (and some data files).

Zeroth printing (revision -70), 08 August 2017

SHORT
Internet

PREFACE

With this issue we have terminated the section "Short Notes."
. . . It has never been "crystal clear" why a Contribution cannot be short,
just as it has occasionally been verified in these pages
that a Short Note might be long.

— ROBERT A. SHORT, *IEEE Transactions on Computers* (1973)

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, 3, and 4A were at the time of their first printings. And alas, those carefully-checked volumes 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 the text both interesting and authoritative, as far as it goes. But the field is vast; I cannot hope to have surrounded it enough to corral it completely. So I beg you to let me know about any deficiencies that you discover.

To put the material in context, this portion of fascicle 5 previews Section 7.2.2.1 of *The Art of Computer Programming*, entitled "Dancing links." It develops an important data structure technique that is suitable for *backtrack programming*, which is the main focus of Section 7.2.2. Several subsections (7.2.2.2, 7.2.2.3, etc.) will follow.

*   *   *

The explosion of research in combinatorial algorithms since the 1970s has meant that I cannot hope to be aware of all the important ideas in this field. I've tried my best to get the story right, yet I fear that in many respects I'm woefully ignorant. So I beg expert readers to steer me in appropriate directions.

Please look, for example, at the exercises that I've classed as research problems (rated with difficulty level 46 or higher), namely exercises 282, 363, . . . ; I've also implicitly mentioned or posed additional unsolved questions in the answers to exercises 13, 75, 78, 182, 186, 268, 310, 350, 356, 361, ... Are those problems still open? Please inform me 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 receive 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 deserves to be credited, with respect to the ideas found in exercises 1, 2, 4, 13, 18, 19, 20, 21, 22, 23, 26, 27, 28, 29, 31, 34, 35, 48, 68, 73, 143, 158, 160, 161, 175, 185, 190, 256, 258, 263, 277, 298(d), 303, 306, 307, 308, 310, 318, 322, 340, 341, 342, 343, 344, 347, 348, 349, 352, 353, 355, 358, 361, 362, ... Furthermore I've credited exercises ... to unpublished work of .... Have any of those results ever appeared in print, to your knowledge?

Jellis
Huang
Sicherman
FGbook
Knuth

* * *

Special thanks are due to George Jellis for answering dozens of historical queries, as well as to Wei-Hwa Huang, George Sicherman, and ... for their detailed comments on my early attempts at exposition. And I want to thank numerous other correspondents who have contributed crucial corrections.

* * *

I happily offer 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 do my best to give you immortal glory, by publishing your name in the eventual book:–)

In the preface to Volume 4B I plan to introduce the abbreviation *FGbook* for my book *Selected Papers on Fun & Games* (Stanford: CSLI Publications, 2011), because I will be making frequent reference to it in connection with recreational problems.

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.
99 Umbruary 2016

MPR
English words
Internet

Part of the Preface to Volume 4B

During the years that I've been preparing Volume 4, I've often run across basic techniques of probability theory that I would have put into Section 1.2 of Volume 1 if I'd been clairvoyant enough to anticipate them in the 1960s. Finally I realized that I ought to collect most of them together in one place, near the beginning of Volume 4B, because the story of these developments is too interesting to be broken up into little pieces scattered here and there.

Therefore this volume begins with a special section entitled "Mathematical Preliminaries Redux," and future sections use the abbreviation 'MPR' to refer to its equations and its exercises.

*   *   *

Several exercises involve the lists of English words that I've used in preparing examples. You'll need the data from

`http://www-cs-faculty.stanford.edu/~knuth/wordlists.tgz`

if you have the courage to work those exercises.

BARRIS
FIELDS
undoing
doubly linked list
LLINK
RLINK
garbage collection
List head
undeletion
delete

What a dance
do they do
Lordy, how I'm tellin' you!
— HARRY BARRIS, *Mississippi Mud* (1927)

Don't lose your confidence if you slip,
Be grateful for a pleasant trip,
And pick yourself up, dust yourself off, start all over again.
— DOROTHY FIELDS, *Pick Yourself Up* (1936)

7.2.2.1. Dancing links. One of the chief characteristics of backtrack algorithms is the fact that they usually need to *undo* everything that they *do* to their data structures. In this section we'll study some extremely simple link-manipulation techniques that modify and unmodify the structures with ease. We'll also see that these ideas have many, many practical applications.

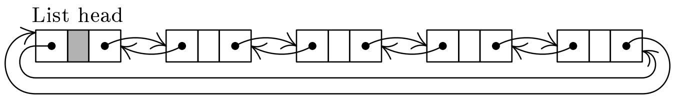
Suppose we have a doubly linked list, in which each node X has a predecessor and successor denoted respectively by LLINK(X) and RLINK(X). Then we know that it's easy to delete X from the list, by setting

$$\texttt{RLINK(LLINK(X))} \leftarrow \texttt{RLINK(X)}, \qquad \texttt{LLINK(RLINK(X))} \leftarrow \texttt{LLINK(X)}. \qquad (1)$$

At this point the conventional wisdom is to recycle node X, making it available for reuse in another list. We might also want to tidy things up by clearing LLINK(X) and RLINK(X), so that stray pointers to nodes that are still active cannot lead to trouble. (See, for example, Eq. 2.2.5–(4), which is the same as (1) except that it also says '$\texttt{AVAIL} \Leftarrow \texttt{X}$'.) By contrast, the dancing-links trick resists any urge to do garbage collection. *In a backtrack application, we're better off leaving* LLINK(X) *and* RLINK(X) *unchanged.* Then we can undo operation (1) by simply setting

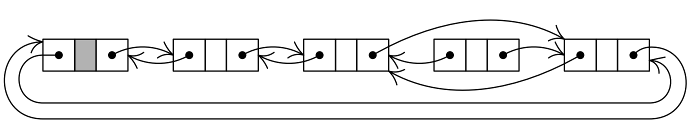
$$\texttt{RLINK(LLINK(X))} \leftarrow \texttt{X}, \qquad \texttt{LLINK(RLINK(X))} \leftarrow \texttt{X}. \qquad (2)$$

For example, we might have a 4-element list, as in 2.2.5–(2):



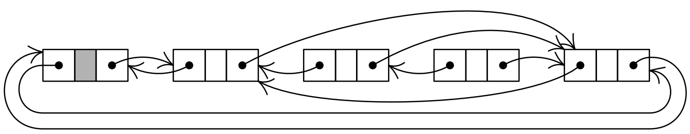
. (3)

If we use (1) to delete the third element, (3) becomes



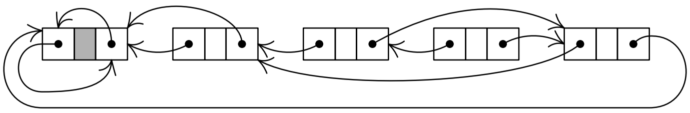
.

And if we now decide to delete the second element also, we get



.

Subsequent deletion of the final element, then the first, will leave us with this:



(4)

undelete
exact covering–
0s and 1s
options
items

The list is now empty, and its links have become rather tangled. (See exercise 1.) But we know that if we proceed to backtrack at this point, using (2) to undelete elements 1, 4, 2, and 3 in that order, we will magically restore the initial state (3). The choreography that underlies the motions of these pointers is fun to watch, and it explains the name "dancing links."

Exact cover problems. We will be seeing many examples where links dance happily and efficiently, as we study more and more examples of backtracking. The beauty of the idea can perhaps be seen most naturally in an important class of problems known as *exact covering*: We're given an $m \times n$ matrix A of 0s and 1s, and the problem is to find a subset of rows whose sum is exactly 1 in every column. For example, consider the 6×7 matrix

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}. \qquad (5)$$

Each row of A corresponds to a subset of a 7-element universe. A moment's thought shows that there's only one way to cover all seven of these columns with disjoint rows, namely by choosing rows 1, 4, and 5. We want to teach a computer how to solve such problems, when there are many, many rows and many columns.

Matrices of 0s and 1s appear frequently in combinatorial problems, and they help us to understand the relations between problems that are essentially the same although they appear to be different (see exercise 5). But inside a computer, we rarely want to represent an exact cover problem explicitly as a two-dimensional array of bits, because the matrix tends to be extremely sparse: There normally are very few 1s. Thus we'll use a different representation, essentially with one node in our data structure for each 1 in the matrix.

Furthermore, we won't even talk about rows and columns! Some of the exact cover problems we deal with already involve concepts that are called "rows" and "columns" in their own areas of application. Instead we will speak of *options* and *items*: Each option is a set of items; and the goal of an exact cover problem is to find disjoint options that cover all the items.

For example, we shall regard (5) as the specification of six options involving seven items. Let's name the items a, b, c, d, e, f, g; then the options are

$$`c\ e'; \quad `a\ d\ g'; \quad `b\ c\ f'; \quad `a\ d\ f'; \quad `b\ g'; \quad `d\ e\ g'. \qquad (6)$$

The first, fourth, and fifth options give us each item exactly once.

covering an item
active list of items

One of the nicest things about exact cover problems is that every tentative choice we make leaves us with a residual exact cover problem that is smaller — often substantially smaller. For example, suppose we try to cover item a in (6) by choosing the option 'a d g': The residual problem has only two options,

$$\text{`}c\ e\text{'} \qquad \text{and} \qquad \text{`}b\ c\ f\text{'}, \tag{7}$$

because the other four involve the already-covered items. Now it's easy to see that (7) has no solution; therefore we can *remove* option 'a d g' from (6). That leaves us with only one option for item a, namely 'a d f'. And its residual,

$$\text{`}c\ e\text{'} \qquad \text{and} \qquad \text{`}b\ g\text{'}, \tag{8}$$

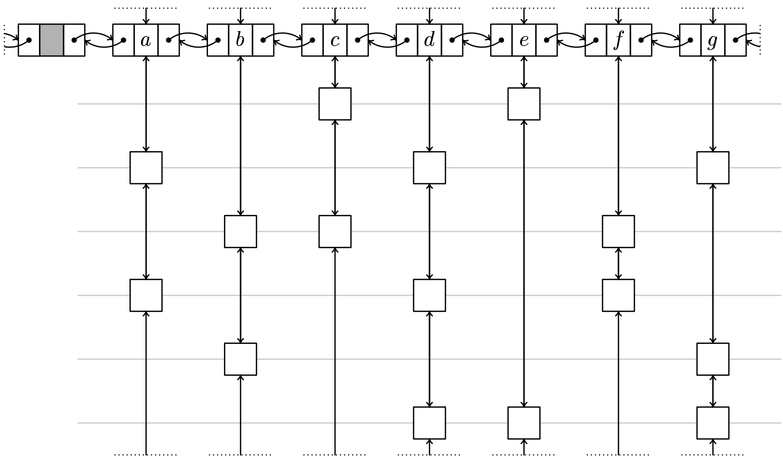
gives us the solution we were looking for.

Thus we're led naturally to a recursive algorithm that's based on the primitive operation of "covering an item": *To cover item i, we delete all of the options that involve i, from our database of currently active options, and we also delete i from the list of items that need to be covered.* The algorithm is simply this:

- Select an item i that needs to be covered; but terminate successfully if none are left (we've found a solution).
- If no active options involve i, terminate unsuccessfully (there's no solution). Otherwise cover item i. (9)
- For each just-deleted option O that involves i, cover each item $j \neq i$ in O, and solve the residual problem.

(Everything that's covered must later be uncovered, of course, as we'll see.)

Interesting details arise when we flesh out this algorithm and look at appropriate low-level mechanisms. There's a doubly linked "horizontal" list of all items that need to be covered; and each item also has its own "vertical" list of all the active options that involve it. For example, the data structures for (6) are:



(10)

(In this diagram, doubly linked pointers "wrap around" at the dotted lines.) The horizontal list has `LLINK` and `RLINK` pointers; the vertical lists have `ULINK` and `DLINK`. Nodes of each vertical list also point to their list header via `TOP` fields.

spacer nodes

The top row of diagram (10) shows the initial state of the horizontal item list and its associated vertical headers. The other rows illustrate the six options of (6), which are represented by sixteen nodes within the vertical lists. Those options implicitly form horizontal lists, indicated by light gray lines; but their nodes *don't* need to be linked together with pointers, because the option lists don't change. We can therefore save time and space by allocating them sequentially. On the other hand, our algorithm does require an ability to traverse each option cyclically, in both directions. Therefore we insert *spacer nodes* between options. A spacer node x is identified by the condition $\mathtt{TOP}(x) \le 0$; it also has

$$\begin{aligned} \mathtt{ULINK}(x) &= \text{address of first node in the option before } x; \\ \mathtt{DLINK}(x) &= \text{address of last node in the option after } x. \end{aligned} \tag{11}$$

These conventions lead to the internal memory layout shown in Table 1. First come the records for individual items; those records have `NAME`, `LLINK`, and `RLINK` fields, where `NAME` is used in printouts. Then come the nodes, which have `TOP`, `ULINK`, and `DLINK` fields. The `TOP` field is, however, called `LEN` in the nodes that serve as item headers, because Algorithm D uses those fields to store the lengths of the item lists. Nodes 8, 11, 15, 19, 23, 26, and 30 in this example are the spacers. Fields marked '—' are unused.

Table 1

THE INITIAL CONTENTS OF MEMORY CORRESPONDING TO (10)

i:	0	1	2	3	4	5	6	7
NAME(i):	—	a	b	c	d	e	f	g
LLINK(i):	7	0	1	2	3	4	5	6
RLINK(i):	1	2	3	4	5	6	7	0
x:	0	1	2	3	4	5	6	7
LEN(x):	—	2	2	2	3	2	2	3
ULINK(x):	—	20	24	17	27	28	22	29
DLINK(x):	—	12	16	9	13	10	18	14
x:	8	9	10	11	12	13	14	15
TOP(x):	0	3	5	−1	1	4	7	−2
ULINK(x):	—	3	5	9	1	4	7	12
DLINK(x):	10	17	28	14	20	21	25	18
x:	16	17	18	19	20	21	22	23
TOP(x):	2	3	6	−3	1	4	6	−4
ULINK(x):	2	9	6	16	12	13	18	20
DLINK(x):	24	3	22	22	1	27	6	25
x:	24	25	26	27	28	29	30	
TOP(x):	2	7	−5	4	5	7	−6	
ULINK(x):	16	14	24	21	10	25	27	
DLINK(x):	2	29	29	4	5	7	—	

OK, we're ready now to spell out precisely what goes on at memory level when our algorithm wants to cover a given item i:

$$\mathrm{cover}(i) = \begin{cases} \text{Set } p \leftarrow \mathtt{DLINK}(i)\text{. (Here } p, l, \text{ and } r \text{ are local variables.)} \\ \text{While } p \ne i, \text{ hide}(p), \text{ then set } p \leftarrow \mathtt{DLINK}(p) \text{ and repeat.} \\ \text{Set } l \leftarrow \mathtt{LLINK}(i),\ r \leftarrow \mathtt{RLINK}(i), \\ \qquad \mathtt{RLINK}(l) \leftarrow r,\ \mathtt{LLINK}(r) \leftarrow l. \end{cases} \tag{12}$$

uncovering an item
hiding an option
unhiding an option
in situ changes
backtracking
MRV

$$\text{hide}(p) = \begin{cases} \text{Set } q \leftarrow p+1, \text{ and repeat the following until } q = p\text{:} \\ \quad \text{Set } x \leftarrow \texttt{TOP}(q),\ u \leftarrow \texttt{ULINK}(q),\ d \leftarrow \texttt{DLINK}(q); \\ \quad \text{if } x \le 0, \text{ set } q \leftarrow u; \\ \quad \text{otherwise set } \texttt{DLINK}(u) \leftarrow d,\ \texttt{ULINK}(d) \leftarrow u, \\ \qquad \texttt{LEN}(x) \leftarrow \texttt{LEN}(x) - 1,\ q \leftarrow q+1. \end{cases} \tag{13}$$

And — here's the point — those operations can readily be undone:

$$\text{uncover}(i) = \begin{cases} \text{Set } l \leftarrow \texttt{LLINK}(i),\ r \leftarrow \texttt{RLINK}(i), \\ \qquad \texttt{RLINK}(l) \leftarrow i,\ \texttt{LLINK}(r) \leftarrow i. \\ \text{Set } p \leftarrow \texttt{ULINK}(i). \\ \text{While } p \ne i, \text{ unhide}(p), \text{ then set } p \leftarrow \texttt{ULINK}(p) \text{ and repeat.} \end{cases} \tag{14}$$

$$\text{unhide}(p) = \begin{cases} \text{Set } q \leftarrow p-1, \text{ and repeat the following until } q = p\text{:} \\ \quad \text{Set } x \leftarrow \texttt{TOP}(q),\ u \leftarrow \texttt{ULINK}(q),\ d \leftarrow \texttt{DLINK}(q); \\ \quad \text{if } x \le 0, \text{ set } q \leftarrow d; \\ \quad \text{otherwise set } \texttt{DLINK}(u) \leftarrow q,\ \texttt{ULINK}(d) \leftarrow q, \\ \qquad \texttt{LEN}(x) \leftarrow \texttt{LEN}(x) + 1,\ q \leftarrow q-1. \end{cases} \tag{15}$$

We're careful here to do everything backwards, using operation (2) inside (14) and (15) to undelete in precisely the reverse order of the way we'd used operation (1) inside (12) and (13) to delete. Furthermore, we're able to do this in place, without copying, by walking through the data structure at the same time as we're modifying it.

Algorithm D (*Exact cover via dancing links*). This algorithm visits all solutions to a given exact cover problem, using the data structures just described. It also maintains a list $x_0, x_1, \ldots, x_T$ of node pointers for backtracking, where T is large enough to accommodate one entry for each option in a solution.

D1. [Initialize.] Set the problem up in memory, as in Table 1 (see exercise 8). Also set N to the number of items, Z to the last spacer address, and $l \leftarrow 0$.

D2. [Enter level l.] If $\texttt{RLINK}(0) = 0$ (hence all items have been covered), visit the solution that is specified by $x_0 x_1 \ldots x_{l-1}$ and go to D8. (See exercise 12.)

D3. [Choose i.] At this point the items $i_1, \ldots, i_t$ still need to be covered, where $i_1 = \texttt{RLINK}(0)$, $i_{j+1} = \texttt{RLINK}(i_j)$, $\texttt{RLINK}(i_t) = 0$. Choose one of them, and call it i. (The MRV heuristic of exercise 9 often works well in practice.)

D4. [Cover i.] Cover item i using (12), and set $x_l \leftarrow \texttt{DLINK}(i)$.

D5. [Try x_l.] If $x_l = i$, go to D7 (we've tried all options for i). Otherwise set $p \leftarrow x_l + 1$, and do the following while $p \ne x_l$: Set $j \leftarrow \texttt{TOP}(p)$; if $j \le 0$, set $p \leftarrow \texttt{ULINK}(p)$; otherwise cover($j$) and set $p \leftarrow p+1$. (This covers the items $\ne i$ in the option that contains p.) Set $l \leftarrow l+1$ and return to D2.

D6. [Try again.] Set $p \leftarrow x_l - 1$, and do the following while $p \ne x_l$: Set $j \leftarrow \texttt{TOP}(p)$; if $j \le 0$, set $p \leftarrow \texttt{DLINK}(p)$; otherwise uncover($j$) and set $p \leftarrow p-1$. (This uncovers the items $\ne i$ in the option that contains p, using the reverse of the order in D5.) Set $i \leftarrow \texttt{TOP}(x_l)$, $x_l \leftarrow \texttt{DLINK}(x_l)$, and return to D5.

D7. [Backtrack.] Uncover item i using (14).

D8. [Leave level l.] Terminate if $l = 0$. Otherwise set $l \leftarrow l-1$ and go to D6. ▌

Langford pairs
symmetry
n queens

The reader is strongly advised to work exercise 10 now — yes, now, really! — in order to experience the dance steps of this instructive algorithm. When the procedure terminates, all of the links will be restored to their original settings.

We're going to see lots of applications of Algorithm D, and similar algorithms, in this section. Let's begin by fulfilling a promise that was made on page 2 of Chapter 7, namely to solve the problem of *Langford pairs* efficiently by means of dancing links.

The task is to put $2n$ numbers $\{1, 1, 2, 2, \ldots, n, n\}$ into $2n$ slots $s_1 s_2 \ldots s_{2n}$, in such a way that exactly i numbers fall between the two occurrences of i. It illustrates exact covering nicely, because we can regard the n values of i and the $2n$ slots s_j as items to be covered. The allowable options for placing the i's are then

$$\text{`}i\ s_j\ s_k\text{'}, \qquad \text{for } 1 \le j < k \le 2n, \quad k = i + j + 1, \quad 1 \le i \le n; \tag{16}$$

for example, when $n = 3$ they're

$$\text{`}1\,s_1\,s_3\text{'}, \text{`}1\,s_2\,s_4\text{'}, \text{`}1\,s_3\,s_5\text{'}, \text{`}1\,s_4\,s_6\text{'}, \text{`}2\,s_1\,s_4\text{'}, \text{`}2\,s_2\,s_5\text{'}, \text{`}2\,s_3\,s_6\text{'}, \text{`}3\,s_1\,s_5\text{'}, \text{`}3\,s_2\,s_6\text{'}. \tag{17}$$

An exact covering of all items is equivalent to placing each pair and filling each slot. Algorithm D quickly determines that (17) has just two solutions,

$$\text{`}3\,s_1\,s_5\text{'}, \text{`}2\,s_3\,s_6\text{'}, \text{`}1\,s_2\,s_4\text{'} \qquad \text{and} \qquad \text{`}3\,s_2\,s_6\text{'}, \text{`}2\,s_1\,s_4\text{'}, \text{`}1\,s_3\,s_5\text{'},$$

corresponding to the placements 3 1 2 1 3 2 and 2 3 1 2 1 3. Notice that those placements are mirror images of each other; exercise 14 shows how to save a factor of 2 and eliminate such symmetry, by omitting some of the options in (16).

With that change, there are exactly 326,721,800 solutions when $n = 16$, and Algorithm D needs about 1.13 trillion memory accesses to visit them all. That's pretty good — it amounts to roughly 3460 mems per solution, as the links whirl.

Of course, we've already looked at a backtrack procedure that's specifically tuned to the Langford problem, namely Algorithm 7.2.2L near the beginning of Section 7.2.2. With the enhancement of exercise 7.2.2–21, that one handles the case $n = 16$ somewhat faster, finishing after about 400 billion mems. But Algorithm D can be pleased that its general-purpose machinery isn't way behind the best custom-tailored method.

Secondary items. Can the classical problem of n queens also be formulated as an exact cover problem? Yes, of course! But the construction isn't quite so obvious. Instead of setting the problem up as we did in 7.2.2–(3), where we chose a queen placement for each row of the board, we shall now allow both rows and columns to participate equally when making the necessary choices.

There are n^2 options for placing queens, and we want exactly one queen in every row and exactly one in every column. Furthermore, we want *at most* one queen in every diagonal. More precisely, if x_{ij} is the binary variable that signifies a queen in row i and column j, we want

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \text{for } 1 \le j \le n; \qquad \sum_{j=1}^{n} x_{ij} = 1 \quad \text{for } 1 \le i \le n; \tag{18}$$

$$\textstyle\sum\{x_{ij} \mid 1 \le i,j \le n,\ i+j=s\} \le 1 \quad \text{for } 1 < s \le 2n; \qquad (19)$$

$$\textstyle\sum\{x_{ij} \mid 1 \le i,j \le n,\ i-j=d\} \le 1 \quad \text{for } -n < d < n. \qquad (20)$$

slack variables
primary
secondary
N_1

The inequalities in (19) and (20) can be changed to equalities by introducing "slack variables" $u_2, \ldots, u_{2n}, v_{-n+1}, \ldots, v_{n-1}$, each of which is 0 or 1:

$$\textstyle\sum\{x_{ij} \mid 1 \le i,j \le n,\ i+j=s\} + u_s = 1 \quad \text{for } 1 < s \le 2n; \qquad (21)$$

$$\textstyle\sum\{x_{ij} \mid 1 \le i,j \le n,\ i-j=d\} + v_d = 1 \quad \text{for } -n < d < n. \qquad (22)$$

Thus we've shown that the problem of n nonattacking queens is equivalent to the problem of finding n^2+4n-2 binary variables x_{ij}, u_s, v_d for which certain subsets of the variables sum to 1, as specified in (18), (21), and (22).

And that is essentially an exact cover problem, whose options correspond to the binary variables and whose items correspond to the subsets. The items are r_i, c_j, a_s, and b_d, representing respectively row i, column j, upward diagonal s, and downward diagonal d. The options are '$r_i\ c_j\ a_{i+j}\ b_{i-j}$' for queen placements, together with trivial options 'a_s' and 'b_d' to take up any slack.

For example, when $n=4$ the n^2 placement options are

$$\begin{array}{llll}
`r_1\ c_1\ a_2\ b_0\text{'}; & `r_2\ c_1\ a_3\ b_1\text{'}; & `r_3\ c_1\ a_4\ b_2\text{'}; & `r_4\ c_1\ a_5\ b_3\text{'}; \\
`r_1\ c_2\ a_3\ b_{-1}\text{'}; & `r_2\ c_2\ a_4\ b_0\text{'}; & `r_3\ c_2\ a_5\ b_1\text{'}; & `r_4\ c_2\ a_6\ b_2\text{'}; \\
`r_1\ c_3\ a_4\ b_{-2}\text{'}; & `r_2\ c_3\ a_5\ b_{-1}\text{'}; & `r_3\ c_3\ a_6\ b_0\text{'}; & `r_4\ c_3\ a_7\ b_1\text{'}; \\
`r_1\ c_4\ a_5\ b_{-3}\text{'}; & `r_2\ c_4\ a_6\ b_{-2}\text{'}; & `r_3\ c_4\ a_7\ b_{-1}\text{'}; & `r_4\ c_4\ a_8\ b_0\text{'};
\end{array} \qquad (23)$$

and the $4n-2$ slack options (which contain just each item each) are

$$`a_2\text{'};\ `a_3\text{'};\ `a_4\text{'};\ `a_5\text{'};\ `a_6\text{'};\ `a_7\text{'};\ `a_8\text{'};\ `b_{-3}\text{'};\ `b_{-2}\text{'};\ `b_{-1}\text{'};\ `b_0\text{'};\ `b_1\text{'};\ `b_2\text{'};\ `b_3\text{'}. \qquad (24)$$

Algorithm D will solve this small problem easily, although its treatment of the slacks is somewhat awkward (see exercise 15).

A closer look shows, however, that a slight change to Algorithm D will allow us to avoid slack options entirely! Let's divide the items of an exact cover problem into two groups: *primary* items, which must be covered *exactly* once, and *secondary* items, which must be covered *at most* once. If we simply modify step D1 so that only the primary items appear in the active list, everything will work like a charm. (Think about it.) In fact, the necessary changes to step D1 already appear in the answer to exercise 8.

Secondary items turn out to be extremely useful in applications. So let's redefine the exact cover problem, taking them into account: Henceforth we shall assume that an exact cover problem involves N distinct items, of which N_1 are primary and $N_2 = N - N_1$ are secondary. It is defined by a family of options, each of which is a subset of the items. *Every option must include at least one primary item.* The task is to find all subsets of the options that (i) contain every primary item exactly once, and (ii) contain every secondary item at most once.

(Options that are purely secondary are excluded from this new definition, because they will never be chosen by Algorithm D as we've refined it. If for some reason you don't like that rule, you can always go back to the idea of slack options. Exercise 18 discusses another interesting alternative.)

organ-pipe order
bitwise operations
diagonals
symmetry
pairwise ordering

The order in which primary items appear in Algorithm D's active list can have a significant effect on the running time, because the implementation of step D3 in exercise 9 selects the *first* item of minimum length. For example, if we consider the primary items of the n queens problem in the natural order $r_1, c_1, r_2, c_2, \ldots, r_n, c_n$, queens tend to be placed at the top and left before we try to place them at the bottom and right. By contrast, if we use the "organ-pipe order" $r_{\lfloor n/2\rfloor+1}, c_{\lfloor n/2\rfloor}, r_{\lfloor n/2\rfloor+2}, c_{\lfloor n/2\rfloor-1}, \ldots$, the queens are placed first in the center, where they prune the remaining possibilities more effectively. For example, the time needed to find all 14772512 of the 16-queen placements is 86 Gμ with the natural order, but only 46 Gμ with the organ-pipe order.

These running times can be compared with 9 Gμ, which we obtained using efficient bitwise operations with Algorithm 7.2.2W. Although that algorithm was specially tuned, the general-purpose dancing links technique runs only five times slower. Furthermore, the setup we've used here allows us to solve other problems that wouldn't be anywhere near as easy with ordinary backtrack. For example, we can limit the solutions to those that contain queens on each of the $2+4l$ longest diagonals, simply by regarding a_s and b_d as primary items instead of secondary, for $|n+1-s|\le l$ and $|d|\le l$. (There are 18048 such solutions when $n=16$ and $l=4$, found with organ-pipe order in 3 Gμ.)

We can also use secondary items to remove symmetry, so that most of the solutions are found only once instead of eight times (see exercises 21 and 22). The central idea is to use a *pairwise ordering* trick that works also in many other situations. Consider the following family of $2m$ options:

$$\alpha_j = \text{`}a\ x_0\ \ldots\ x_{j-1}\text{'} \quad\text{and}\quad \beta_j = \text{`}b\ x_j\text{'} \qquad \text{for } 0\le j<m, \tag{25}$$

where a and b are primary while $x_0, x_1, \ldots, x_{m-1}$ are secondary. For example, when $m=4$ the options are

$$\begin{array}{ll} \alpha_0 = \text{`}a\text{'}; & \beta_0 = \text{`}b\ x_0\text{'}; \\ \alpha_1 = \text{`}a\ x_0\text{'}; & \beta_1 = \text{`}b\ x_1\text{'}; \\ \alpha_2 = \text{`}a\ x_0\ x_1\text{'}; & \beta_2 = \text{`}b\ x_2\text{'}; \\ \alpha_3 = \text{`}a\ x_0\ x_1\ x_2\text{'}; & \beta_3 = \text{`}b\ x_3\text{'}. \end{array}$$

It's not hard to see that there are exactly $\binom{m+1}{2}$ ways to cover a and b, namely to choose α_j and β_k with $0\le j\le k<m$. For if we choose α_j, the secondary items x_0 through x_{j-1} knock out the options $\beta_0, \ldots, \beta_{j-1}$.

This construction involves a total of $\binom{m+1}{2}$ entries in the α options and $2m$ entries in the β options. But exercise 19 shows that it's possible to achieve pairwise ordering with only $O(m\log m)$ entries in both α's and β's. For example, when $m=4$ it produces the following elegant pattern:

$$\begin{array}{ll} \alpha_0 = \text{`}a\text{'}; & \beta_0 = \text{`}b\ y_1\ y_2\text{'}; \\ \alpha_1 = \text{`}a\ y_1\text{'}; & \beta_1 = \text{`}b\ y_2\text{'}; \\ \alpha_2 = \text{`}a\ y_2\text{'}; & \beta_2 = \text{`}b\ y_3\text{'}; \\ \alpha_3 = \text{`}a\ y_3\ y_2\text{'}; & \beta_3 = \text{`}b\text{'}. \end{array} \tag{26}$$

Progress reports. Many of the applications of Algorithm D take a long time, especially when we're using it to solve a tough problem that is breaking new ground. So we don't want to just start it up and wait with our fingers crossed, hoping that it will finish soon. We really want to watch it in action and see how it's doing. How many more hours will it probably run? Is it almost half done?

A simple amendment of step D2 will alleviate such worries. At the beginning of that step, as we enter a new node of the search tree, we can test whether the accumulated running time T has passed a certain threshold Θ, which is initially set to Δ. If $T \ge \Theta$, we print a progress report and set $\Theta \leftarrow \Theta + \Delta$. (Thus if $\Delta = \infty$, we get no reports; if $\Delta = 0$, we see a report at each node.)

The author's experimental program measures time in mems, so that he can obtain machine-independent results; and he often takes $\Delta = 10\,\mathrm{G}\mu$. His program has two main ways to show progress, namely a long form and a short form. The long form gives full details about the current state of the search, based on exercise 11. For example, here's the first progress report that it displays when finding all solutions to the 16 queens problem as described above:

```
Current state (level 15):
 r8 c3 ab ba (4 of 16)
 c8 a8 bn r0 (1 of 13)
 r7 cb ai bj (7 of 10)
 r6 c4 aa bd (2 of 7)
   ...
 3480159 solutions, 10000000071 mems so far.
```

(The computer's internal encoding for items is different from the conventions we have used; for example, '`r8 c3 ab ba`' stands for what we called '$r_9\ c_4\ a_{13}\ b_5$'. The first choice at level 0 was to cover item `r8`, meaning to put a queen into row 9. The fourth of 16 options for that item has placed it in column 4. Then at level 1 we tried the first of 13 ways to cover item `c8`, meaning to put a queen into column 9. And so on. At each level, the leftmost item shown for the option being tried is the one that was chosen in step D3 for branching.)

That's the long form. The short form, which is the default, produces just one line for each state report:

```
10000000071mu: 3480159 sols, 4g 1d 7a 27 36 24 23 13 12 12 22 12 ... .19048
20000000111mu: 6604373 sols, 7g cd 6a 88 36 35 44 44 24 11 12 22 .43074
30000000052mu: 9487419 sols, bg cd 9a 68 37 35 24 13 12 12 .68205
40000000586mu: 12890124 sols, fg 6d aa 68 46 35 23 33 23 .90370
Altogether 14772512 solutions, 62236+45565990457 mems, 178259509 nodes.
```

Two-character codes are used to indicate the current position in the tree; for example, '`4g`' means branch 4 of 16, then '`1d`' means branch 1 of 13, etc. By watching these steadily increasing codes — it's fun! — we can monitor the action.

Each line in the short form ends with an estimate of how much of the tree has been examined, assuming that the search tree structure is fairly consistent. For instance, '`.19043`' means that we're roughly 19% done. If we're currently working at level l on choice c_l of t_l, this number is computed by the formula

$$\frac{c_0-1}{t_0}+\frac{c_1-1}{t_0t_1}+\cdots+\frac{c_l-1}{t_0t_1\ldots t_l}+\frac{1}{2t_0t_1\ldots t_l}. \qquad (27)$$

node
search tree
running time
threshold
progress report
author
mems
queens problem
completion ratio
ratio of completion

sudoku–
soduko: see sudoku
Garns
Number Place
Nikoli
pi, as random
0-origin indexing

Sudoku. A "sudoku square" is a 9×9 array that has been divided into 3×3 boxes and filled with the digits $\{1,2,3,4,5,6,7,8,9\}$ in such a way that

- every row contains each of the digits $\{1,2,3,4,5,6,7,8,9\}$ exactly once;
- every column contains each of the digits $\{1,2,3,4,5,6,7,8,9\}$ exactly once;
- every box contains each of the digits $\{1,2,3,4,5,6,7,8,9\}$ exactly once.

(Since there are nine cells in each row, each column, and each box, the words 'exactly once' can be replaced by 'at least once' or 'at most once', anywhere in this definition.) Here, for example, are three highly symmetrical sudoku squares:

(a)

1	2	3	4	5	6	7	8	9
4	5	6	7	8	9	1	2	3
7	8	9	1	2	3	4	5	6
2	3	4	5	6	7	8	9	1
5	6	7	8	9	1	2	3	4
8	9	1	2	3	4	5	6	7
3	4	5	6	7	8	9	1	2
6	7	8	9	1	2	3	4	5
9	1	2	3	4	5	6	7	8

; (b)

1	2	3	4	5	6	7	8	9
4	5	6	7	8	9	1	2	3
7	8	9	1	2	3	4	5	6
2	3	1	5	6	4	8	9	7
5	6	4	8	9	7	2	3	1
8	9	7	2	3	1	5	6	4
3	1	2	6	4	5	9	7	8
6	4	5	9	7	8	3	1	2
9	7	8	3	1	2	6	4	5

; (c)

1	2	3	4	5	6	7	8	9
5	6	4	8	9	7	2	3	1
9	7	8	3	1	2	6	4	5
6	4	5	9	7	8	3	1	2
7	8	9	1	2	3	4	5	6
2	3	1	5	6	4	8	9	7
8	9	7	2	3	1	5	6	4
3	1	2	6	4	5	9	7	8
4	5	6	7	8	9	1	2	3

. (28)

When the square has been only partially specified, the task of completing it — by filling in the blank cells — often turns out to be a fascinating challenge. Howard Garns used this idea as the basis for a series of puzzles that he called Number Place, first published in *Dell Pencil Puzzles & Word Games* #16 (May 1979), 6. The concept soon spread to Japan, where Nikoli Inc. gave it the name Su Doku (数独, "Unmarried Numbers") in 1984; and eventually it went viral. By the beginning of 2005, major newspapers had begun to feature daily sudoku puzzles. Today, sudoku ranks among the most popular recreations of all time.

Every sudoku puzzle corresponds to an exact cover problem of a particularly nice form. Consider, for example, the following three instances:

(a)

		3		1				
4	1	5					9	
2		6	5			3		
5				8				9
	7		9				3	2
	3	8			4		6	
			2	6		4		3
			3					8
3	2				7	9	5	

; (b)

						3		
1			4					
						1		5
9								
					2	6		
				5	3			
	5		8					
			9				7	
	8	3					4	

; (c)

	3			1				
			4			1		
	5						9	
2						6		4
				3	5			
1								
4			6					
							5	
	9							

. (29)

(The clues in (29a) match the first 32 digits of π; but the clues in (29b) and (29c) disagree with π after awhile.) For convenience, let's number the rows, columns, and boxes from 0 to 8. Then every sudoku square $S = (s_{ij})$ corresponds naturally to the solution of a master exact cover problem whose $9 \cdot 9 \cdot 9 = 729$ options are

$$\text{`}p_{ij}\ r_{ik}\ c_{jk}\ b_{xk}\text{'} \quad \text{for } 0 \le i,j < 9,\ 1 \le k \le 9, \text{ and } x = 3\lfloor i/3 \rfloor + \lfloor j/3 \rfloor, \qquad (30)$$

and whose $4 \cdot 9 \cdot 9 = 324$ items are p_{ij}, r_{ik}, c_{jk}, b_{xk}. The reason is that option (30) is chosen with parameters (i,j,k) if and only if $s_{ij} = k$. Item p_{ij} must be covered by exactly one of the nine options that fill cell (i,j); item r_{ik} must be covered by exactly one of the nine options that put k in row i; ...; item b_{xk} must be covered by exactly one of the nine options that put k in box x. Got it?

My own motive for writing on the subject is partly to justify the appalling number of hours I have squandered solving Sudoku.
— BRIAN HAYES, in *American Scientist* (2006)

naked single
MRV heuristic
Royle
McGuire

To find all sudoku squares that contain a given *partial* specification, we simply remove all of the items p_{ij}, r_{ik}, c_{jk}, b_{xk} that are already covered, as well as all of the options that involve any of those items. For example, (29a) leads to an exact cover problem with $4 \cdot (81 - 32) = 196$ items p_{00}, p_{01}, p_{03}, …, r_{02}, r_{04}, r_{05}, …, c_{01}, c_{06}, c_{07}, …, b_{07}, b_{08}, b_{09}, …; it has 146 options, beginning with 'p_{00} r_{07} c_{07} b_{07}' and ending with 'p_{88} r_{86} c_{86} b_{86}'. These options can be visualized by making a chart that shows every value that hasn't been ruled out:

	0	1	2	3	4	5	6	7	8
0	789	89	**3**	4678	**1**	2689	25678	2478	4567
1	**4**	**1**	**5**	678	237	2368	2678	**9**	67
2	**2**	89	**6**	**5**	479	89	**3**	1478	147
3	**5**	46	124	167	**8**	1236	17	147	**9**
4	16	**7**	14	**9**	5	156	158	**3**	**2**
5	19	**3**	**8**	17	257	**4**	157	**6**	157
6	1789	589	179	**2**	**6**	1589	**4**	17	**3**
7	1679	4569	1479	**3**	459	159	1267	127	**8**
8	**3**	**2**	14	148	4	**7**	**9**	**5**	16

(31)

The active list for item p_{00}, say, has options for values $\{7, 8, 9\}$; the active list for item r_{02} has options for columns $\{5, 6, 7\}$; the active list for item c_{01} has options for rows $\{4, 5, 6, 7\}$; and so on. (Indeed, sudoku experts tend to have charts like this in mind, implicitly or explicitly, as they work.)

Aha! Look at the lonely '5' in the middle! There's only one option for p_{44}; so we can promote that '5' to '**5**'. Hence we can also erase the other '5's that appear in row 4, column 4, or box 4. This operation is called "forcing a naked single."

And there's *another* naked single in cell $(8, 4)$. Promoting this one from '4' to '**4**' produces others in cells $(7, 4)$ and $(8, 2)$. Indeed, if the items p_{ij} have been placed first in step D1, Algorithm D will follow a merry path of forced moves that lead immediately to a complete solution of (29a), *entirely* via naked singles.

Of course sudoku puzzles aren't always this easy. For example, (29b) has only 17 clues, not 32; that makes naked singles less likely. (Puzzle (29b) comes from Gordon Royle's online collection of approximately 50,000 17-clue sudokus — all of which are essentially different, and all of which have a unique completion despite the paucity of clues. Royle's collection appears to be nearly complete: Whenever a sudoku fanatic encounters a 17-clue puzzle nowadays, that puzzle almost invariably turns out to be equivalent to one in Royle's list.)

A massive computer calculation, supervised by Gary McGuire and completed in 2012, has shown that *every uniquely solvable sudoku puzzle must* *contain at least 17 clues.* We will see in Section 7.2.3 that exactly 5,472,730,538 nonisomorphic sudoku squares exist. McGuire's program examined each of them, and showed that comparatively few 16-clue subsets could possibly characterize it. About 16,000 subsets typically survived this initial screening, but they too were shown to fail. All this was determined in roughly 3.6 seconds per sudoku square, thanks to nontrivial and highly optimized bitwise algorithms. [See G. McGuire, B. Tugemann, and G. Civario, *Experimental Mathematics* **23** (2014), 190–217.]

bitwise
McGuire
Tugemann
Civario
naked singles
hidden single

The 17 clues of puzzle (29b) produce the following chart analogous to (31):

	0	1	2	3	4	5	6	7	8
0	245678	24679	2456789	12567	126789	156789	**3**	2689	246789
1	**1**	23679	256789	**4**	236789	56789	2789	2689	26789
2	234678	234679	246789	2367	236789	6789	**1**	2689	**5**
3	**9**	123467	1245678	167	14678	14678	24578	12358	123478
4	34578	1347	14578	17	14789	**2**	**6**	13589	134789
5	24678	12467	124678	167	**5**	**3**	24789	1289	124789
6	2467	**5**	124679	**8**	123467	1467	29	12369	12369
7	246	1246	1246	**9**	12346	1456	258	**7**	12368
8	267	**8**	**3**	12567	1267	1567	259	**4**	1269

(32)

This one has 307 options remaining — more than twice as many as before. Also, as we might have guessed, it has no naked singles. But it still reveals forced moves, if we look more closely! For example, column 3 contains only one instance of '3'; we can promote it to '3', and kill all of the other '3's in row 2 and box 1. This operation is called "forcing a hidden single."

Similarly, box 2 in (32) contains only one instance of '4'; and two other hidden singles are also present (see exercise 63). These forced moves cause other hidden singles to appear, and naked singles also arise soon. But after 16 forced promotions have been made, the low-hanging fruit is all gone:

	0	1	2	3	4	5	6	7	8
0	**5**	679	678	**2**	1678	16789	**3**	689	**4**
1	**1**	**3**	2678	**4**	678	**5**	78	2689	2679
2	24678	24679	24678	**3**	678	6789	**1**	2689	**5**
3	**9**	12467	1245678	167	14678	14678	478	12358	1237
4	**3**	147	14578	17	**9**	**2**	**6**	158	17
5	24678	12467	124678	167	**5**	**3**	478	1289	1279
6	467	**5**	**9**	**8**	1467	1467	**2**	136	136
7	246	1246	1246	**9**	**3**	146	**5**	**7**	**8**
8	67	**8**	**3**	**5**	**2**	167	**9**	**4**	16

(33)

Algorithm D readily deduces (33) from (32), because it sees naked singles and hidden singles whenever an item p_{ij} or r_{ik} or c_{jk} or b_{xk} has only one remaining option, and because its data structures change easily as the links dance. But when state (33) is reached, the algorithm resorts to two-way branching, in this case looking first at case '7' of p_{16}, then backtracking later to consider case '8'.

naked pair
author
secondary

A human sudoku expert would actually glance at (33) and notice that there's a more intelligent way to proceed, because (33) contains a "naked pair": Cells $(4,3)$ and $(4,8)$ both contain the same two choices; hence we're allowed to delete '1" and '7' wherever they appear elsewhere in row 4; and this will produce a naked '4' in column 1. Exercise 65 explores such higher-order deductions in detail.

Fancy logic that involves pairs and triples might well be preferable for earthlings, but simple backtracking works just fine for machines. In fact, Algorithm D finds the solution to (29b) after exploring a search tree with just 88 nodes, the first 16 of which led it directly to (33). (It spends about 250 kilomems initializing the data structures in step D1, then 50 more kilomems to complete the task. Much more time would have been needed if it had tried to look for complicated patterns in step D3.) Here's the solution that it discovers:

```
c33 b13 p23 r23 (1 of 1)
r13 c13 b03 p11 (1 of 1)
  .   .   .
c42 b72 p84 r82 (1 of 1)
p16 r18 c68 b28 (2 of 2)
b27 p18 r17 c87 (1 of 1)
  .   .   .
p85 r81 c51 b71 (1 of 1)
```

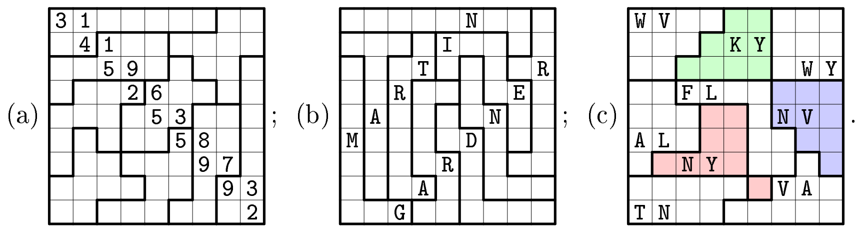
After it selects the correct value for column 6 of row 1, the rest is forced.

The dancing links method actually cruises to victory with amazing speed, on every known sudoku puzzle. Among several dozen typical specimens — seen by the author since 2005 in newspapers, magazines, books, and webpages from around the world, and subsequently presented to Algorithm D — roughly 70% were found to be solvable entirely by forced moves based on naked or hidden singles, even though many of those puzzles had been rated 'diabolical' or 'fiendish' or 'torturous'! Only 10% of them led to a search tree exceeding 100 nodes, and none of the trees had more than 281 nodes.

It's interesting to consider what happens when the algorithm is weakened, so that its forced moves come only from naked singles, which are the easiest deductions for people to make. Suppose we classify the items r_{ik}, c_{jk}, and b_{xk} as *secondary*, leaving only p_{ij} as primary. (In other words, the specification will require *at most* one occurrences of each value k, in every row, every column, and every box, but it won't explicitly insist that every k should be covered.) The search tree for puzzle (29b) then grows to a whopping 41,876 nodes.

Finally, what about puzzle (29c)? That one has only 16 clues, so we know that it cannot have a unique solution. But those 16 clues specify only seven of the nine digits; they give us no way to distinguish 7 from 8. Algorithm D deduces, with a 127-node search tree, that only two solutions exist. (Of course those two are essentially the same; they're obtainable from each other by swapping 7 $\leftrightarrow$ 8.)

Puzzlists have invented many intriguing variations on the traditional sudoku challenge, several of which are discussed in the exercises below. Among the best are “jigsaw sudoku puzzles” (also known as “geometric sudoku,” “polyomino sudoku,” “squiggly sudoku,” etc.), where the boxes have different shapes instead of simply being 3×3 subsquares. Consider, for example,



(34)

In puzzle (34a), which the author designed in 2017 with the help of Bob Harris, one can see for instance that there are only two places to put a ‘4’ in the top row, because of the ‘4’ in the next row. This puzzle is an exact cover problem just like (30), except that x is now a more complicated function of i and j. Similarly, Harris’s classic puzzle (34b) [*Mathematical Wizardry for a Gardner* (2009), 55–57] asks us to put the letters $\{\mathtt{G},\mathtt{R},\mathtt{A},\mathtt{N},\mathtt{D},\mathtt{T},\mathtt{I},\mathtt{M},\mathtt{E}\}$ into each row, column, and irregularly shaped box. Again we use (30), but with the values of k running through letters instead of digits. [*Hint:* Cell $(0,2)$ must contain ‘A’, because column 2 needs an ‘A’ somewhere.] Puzzle (34c), The United States Jigsaw Sudoku, is a masterpiece designed and posted online by Thomas Snyder in 2006. It brilliantly uses boxes in the shapes of West Virginia, Kentucky, Wyoming, Alabama, Florida, Nevada, Tennessee, New York (including Long Island), and Virginia — and its clues are postal codes! (See exercise 70.)

Jigsaw sudoku was invented by Mark Thompson, who begin to publish such puzzles in 1996 [*GAMES World of Puzzles* **2**, 4 (July 1996), ??, ??] under the name Latin Squares. At that time he had not yet heard about sudoku; one of the advantages of his puzzles over normal sudoku was the fact that they can be of any size, not necessarily 9×9. Thompson’s first examples were 6×6.

The *solutions* to puzzles of this kind actually have an interesting prehistory: Walter Behrens, a pioneer in the applications of mathematics to agriculture, wrote an influential paper in 1956 that proposed using such patterns in empirical studies of crops that have been treated with various fertilizers [*Zeitschrift für landwirtschaftliches Versuchs- und Untersuchungswesen* **2** (1956), 176–193]. He presented dozens of designs, ranging from 4×4 to 10×10, including

(a)

1	5	4	2	3
2	4	3	5	1
3	1	2	4	5
4	3	5	1	2
5	2	1	3	4

; (b)

1	3	8	7	9	5	2
2	6	9	5	3	8	4
3	8	2	4	6	7	1
4	7	6	9	1	3	8
5	1	3	8	7	6	9
6	9	7	1	4	2	6
7	5	4	2	8	9	6
8	2	1	3	5	4	7
9	4	5	6	2	1	3

; (c)

1	5	8	6	4	3	9	7	2
2	6	9	5	7	8	1	3	4
3	7	4	1	9	2	5	6	8
4	9	2	8	6	7	3	1	5
5	3	7	4	2	1	6	8	9
6	8	1	3	5	9	2	4	7
7	2	6	9	1	4	8	5	3
8	4	5	2	3	6	7	9	1
9	1	3	7	8	5	4	2	6

. (35)

jigsaw sudoku puzzles
geometric sudoku
polyomino sudoku
squiggly sudoku
author
Harris
pi as src
Gardner
GRAND TIME puzzle
United States Jigsaw Sudoku
Snyder
Thompson
Latin Squares
Behrens
agriculture

Notice that Behrens's (35b) is actually 9×7, so its rows don't exhibit all 9 possibilities. He required only that no treatment number be repeated in any row or column. Notice also that his (35c) is actually a normal sudoku arrangement; this is the earliest known publication of what is now called a sudoku solution. Following a suggestion of F. Ragallen, Behrens called these designs "gerechte" ("equitable") latin squares or latin rectangles, because they assign neighborhood groupings to tracts of land that have been subjected to all n treatments.

All of his designs were partitions of rectangles into connected regions, each with n square cells. We'll see next that *that* idea actually turns out to have its own distinguished history of fascinating combinatorial patterns and recreations.

Polyominoes. A rookwise-connected region of n square cells is called an *n-omino*, following a suggestion by S. W. Golomb [*AMM* **61** (1954), 675–682]. When $n = 1, 2, 3, \ldots$, that gives us monominoes, dominoes, trominoes, tetrominoes, pentominoes, hexominoes, and so on; and when n is unspecified, such regions are called *polyominoes*.

We've already encountered small polyominoes, together with their relation to exact covering, in 7.1.4–(130). It's clear that a domino has only one possible shape. But there are two distinct species of trominoes, one of which is "straight" (1×3) and the other is "bent," occupying three cells of a 2×2 square. Similarly, the tetrominoes can be classified into five distinct types. (Can you draw all five, before looking at exercise 257? Tetris® players will have no trouble doing this.)

The most piquant polyominoes, however, are almost certainly the *pentominoes*, of which there are twelve. These twelve shapes have become the personal friends of millions of people, because they can be put together in so many elegant ways. Sets of pentominoes, made from finely crafted hardwoods or from brilliantly colored plastic, are readily available at reasonable cost. Every home really ought to have at least one such set — even though "virtual" pentominoes can easily be manipulated in computer apps — because there's no substitute for the strangely fascinating tactile experience of arranging these delightful physical objects by hand. Furthermore, we'll see that pentominoes have much to teach us about combinatorial computing.

If mounted on cardboard, [these pieces]
will form a source of perpetual amusement in the home.

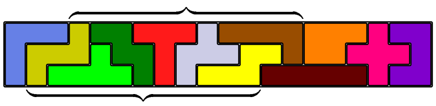
— HENRY E. DUDENEY, *The Canterbury Puzzles* (1907)

Which English nouns ending in -o *pluralize with* -s *and which with* -es*?*
If the word is still felt as somewhat alien, it takes -s,
while if it has been fully naturalized into English, it takes -es.
Thus, echoes, potatoes, tomatoes, dingoes, embargoes, *etc.*,
whereas Italian musical terms are altos, bassos, cantos, pianos, solos, *etc.*,
and there are Spanish words like tangos, armadillos, *etc.*
I once held a trademark on 'Pentomino(-es)', but I now prefer
to let these words be my contribution to the language as public domain.

— SOLOMON W. GOLOMB, letter to Donald Knuth (16 February 1994)

Behrens
sudoku solution
Ragallen
gerechte
Polyominoes
n-omino
Golomb
monominoes
dominoes
trominoes
tetrominoes
pentominoes
hexominoes
polyominoes
straight
bent
Tetris
DUDENEY
trademark
GOLOMB
Knuth

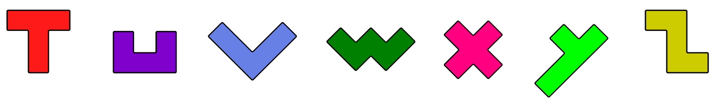
One of the first things we might try to do with twelve pieces of 5 cells each is to pack them into a rectangular box, either 6×10 or 5×12 or 4×15 or 3×20. The first three tasks are fairly easy; but a 3×20 box presents more of a challenge. Golomb posed this question in his article of 1954, without providing any answer. At that time he was unaware that Frans Hansson had already given a solution many years earlier, in an obscure publication called *The Problemist: Fairy Chess Supplement* **2**, 12 and 13 (June and August, 1935), problem 1844:



(36)

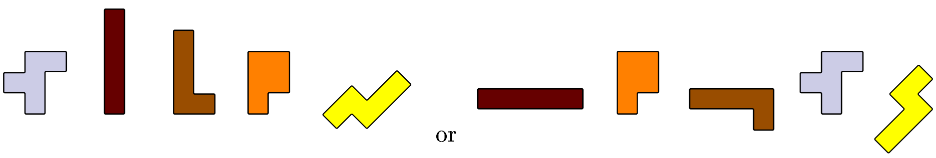
Hansson also observed that the bracketed pieces "may also be rotated through two right angles, to give the only other possible solution."

This problem, and many others of a similar kind, can be formulated nicely in terms of exact covering. But before we do this, we need *names* for the individual pentomino shapes. Everybody agrees that seven of the pentominoes should be named after seven consecutive letters of the alphabet:



T U V W X Y Z

But two different systems of nomenclature have been proposed for the other five:



or

F I L P N O P Q R S

(S. W. Golomb) (J. H. Conway)

where Golomb likes to think of the word 'Filipino' while Conway prefers to map the twelve pentominoes onto twelve consecutive letters. Conway's scheme tends to work better in computer programs, so we'll use it here.

The task of 3×20 pentomino packing consists of arranging pentominoes in such a way that every piece name $\{\texttt{O}, \texttt{P}, \ldots, \texttt{Z}\}$ is covered exactly once, and so is every cell ij for $0 \le i < 3$ and $0 \le j < 20$. Thus there are $12 + 3 \cdot 20 = 72$ items; and there's an option for each way to place an individual pentomino, namely

```
'O 00 01 02 03 04'
        ...
'O 2f 2g 2h 2i 2j'
'P 00 01 02 10 11'              (37)
        ...
'Z 0j 1h 1i 1j 2h'
```

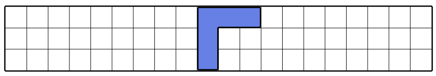
if we extend hexadecimal notation so that the "digits" $(\texttt{a}, \texttt{b}, \ldots, \texttt{j})$ represent $(10, 11, \ldots, 19)$. In this list, pieces $(\texttt{O}, \texttt{P}, \ldots, \texttt{Z})$ contribute respectively (48, 220,

Golomb
Hansson
Golomb
Conway
pentominoes, names of
packing
hexadecimal notation

136, 144, 136, 72, 110, 72, 72, 18, 136, 72) options, making 1236 altogether. Exercise 240 explains how to generate all of the options for problems like this.

When Algorithm D is applied to (37), it finds eight solutions, because each of Hansson's arrangements is obtained with horizontal and/or vertical reflection. We can remove that symmetry by insisting that the V pentomino must appear in its 'Γ-like' orientation, as it does in (36), namely by removing all but 18 of its 72 options. (Do you see why? Think about it.) Without that simplification, an 32,636-node search tree finds 8 solutions in 146 megamems; with it, a 21,803-node search tree finds 2 solutions in 103 megamems.

A closer look shows that we can actually do much better. For example, one of the Γ-like options for V is '`V 09 0a 0b 19 29`', representing


; (38)

but this placement could never be used, because it asks us to pack pentominoes into the 27 cells at V's left. Many of the options for other pieces are similarly unusable, because (like (38)) they isolate a region whose area isn't a multiple of 5.

In fact, if we remove all such options, only 728 of the original 1236 potential placements remain; they include respectively (48, 156, 132, 28, 128, 16, 44, 16, 12, 4, 128, 16) placements of (`O`, `P`, ..., `Z`). That gives us 716 options, when we remove also the 12 surviving placements for V that make it non-'Γ'. When Algorithm D is applied to this reduced set, the search tree for finding all solutions goes down to 1241 nodes, and the running time is only 4.5 megamems.

(There's also a slightly better way to remove the symmetry: Instead of insisting that piece V looks like 'Γ' we can insist that piece X lies in the left half, and that piece Z hasn't been "flipped over." This implies that there are (16, 2, 8) potential placements for (`V`, `X`, `Z`), instead of (4, 4, 16). The resulting search tree has just 1126 nodes, and the running time is 4.0 Mμ.)

Notice that we could have begun with a weaker formulation of this problem: We could merely have asked for pentomino arrangements that use each piece *at most* once, while covering each cell ij exactly once. That would be essentially the same as saying that the piece names $\{\texttt{O}, \texttt{P}, \ldots, \texttt{Z}\}$ are *secondary* items instead of primary. Then the original set of 1236 options in (37) would have led to a search tree with 61,843 nodes, and a runtime of 291 Mμ. Dually, we could have kept the piece names primary but made the cell names secondary; that would have yielded a 1,086,521,913-node tree, with a runtime of 2.94 Tμ! These statistics are curiously *reversed*, however, with respect to the reduced set of 716 options obtained by discarding cases like (38): Then piece names secondary yields 19304 nodes (68 Mμ); cell names secondary yields 11654 nodes (37 Mμ).

In the early days of computing, pentomino problems served as useful benchmarks for combinatorial calculations. Programmers didn't have the luxury of large random-access memory until much later; therefore techniques such as dancing links, in which more than a thousand options are explicitly listed and manipulated, were unthinkable at the time. Instead, the options for each piece were implicitly generated on-the-fly as needed, and there was no incentive to use

Hansson
symmetry removal
remove the symmetry
flipped over
secondary
benchmarks
history

fancy heuristics while backtracking. Each branch of the search was essentially based on the available ways to cover the first cell ij that hadn't yet been occupied.

We can simulate the behavior of those historic methods by running Algorithm D without the MRC heuristic and simply setting $i \leftarrow$ `RLINK(0)` in step D3. An interesting phenomenon now arises: If the cells ij are considered in their natural order — first `00`, then `01`, ..., then `0j`, then `10`, ..., finally `2j` — the search tree has 1.5 billion nodes. (There are 29 ways to cover `00`; if we choose '`00 01 02 03 04 0`' there are 49 ways to cover `05`; and so on.) But if we consider the 20×3 problem instead of 3×20, so that the cells ij for $0 \le i < 20$ and $0 \le j < 3$ are processed in order `00`, `01`, `02`, `10`, ..., `2j`, the search tree has just 71183 nodes, and all eight solutions are found very quickly. (This speedup is mostly due to having a better "focus," which we'll discuss later.) Again we see that a small change in problem setup can have enormous ramifications.

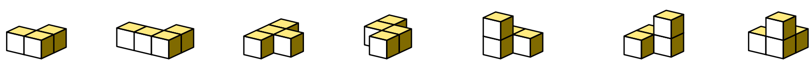
The best of these early programs were highly tuned, written in assembly language with ingenious uses of macro instructions. Memwise, they were therefore superior to Algorithm D on smallish problems. [See J. G. Fletcher, *CACM* **8**, 10 (October 1965), cover and 621–623; N. G. de Bruijn, *FGbook* pages 465–466.] But the MRV heuristic eventually wins, as problems get larger.

Exercises 250–299 discuss some of the many intriguing and instructive problems that arise when we explore the patterns that can be made with pentominoes and hexominoes. Several of these problems are indeed large — beyond the capabilities of today's machines.

Polycubes. And if you think two-dimensional shapes are fun, you'll probably like three dimensions even more! A *polycube* is a solid object formed by taking one or more $1 \times 1 \times 1$ cubes and joining them face-to-face. We call them monocubes, dicubes, tricubes, tetracubes, pentacubes, etc.; but we *don't* call them "n-cubes" when they're made from n unit cubies, because mathematicians have reserved that term for n-dimensional objects.

A new situation arises when $n = 4$. In two dimensions we found it natural to regard the tetromino '' as identical to its mirror image '', because we could simply flip it over. But the tetracube '' is noticeably different from its mirror reflection '', because we can't change one into the other without going into the fourth dimension. Polycubes that differ from their mirror images are called *chiral*, a word coined by Kelvin in 1893 when he studied such molecules.

The simplest polycubes are *cuboids* — also called rectangular parallelepipeds by people who like long names. But things get particularly interesting when we consider noncuboidal shapes. Piet Hein noticed in 1933 that the seven smallest shapes of that kind, namely

 , (39)

1: bent 2: ell 3: tee 4: skew 5: L-twist 6: R-twist 7: claw

can be put together to form a $3 \times 3 \times 3$ cube, and he liked the pieces so much that he called them *Soma*. Notice that the first four pieces are essentially planar, while the other three are inherently three-dimensional. The twists are chiral.

MRC heuristic
focus
assembly language
macro instructions
de Bruijn
MRV heuristic
hexominoes
Polycubes
monocubes
dicubes
tricubes
tetracubes
pentacubes
n-cubes
cubie: A $1 \times 1 \times 1$ cube inside a larger bo
tetromino
flip it over
chiral
Kelvin
cuboids
parallelepipeds
Hein
bent tromino
ell tetromino
tee tetromino
skew tetromino
twist tetracube
claw tetracube
Soma

Martin Gardner wrote about the joys of Soma in *Scientific American* **199**, 3 (September 1958), 182–188, and it soon became wildly popular: More than two million SOMA® cubes were sold in America alone, after Parker Brothers began to market a well-made set with an instruction booklet written by Hein.

Gardner
Parker Brothers
HEIN
Patent
pentominoes
canonical
factoring

> *A minimum number of blocks of simple form are employed. . . .*
> *Experiments and calculations have shown that from the set of seven blocks*
> *it is possible to construct approximately the same number of geometrical*
> *figures as could be constructed from twenty-seven separate cubes.*
> — PIET HEIN, *United Kingdom Patent Specification 420,349* (1934)

The task of packing these seven pieces into a cube is easy to formulate as an exact cover problem, just as we did when packing pentominoes. But this time we have 24 3D-rotations of the pieces to consider, instead of 8 2D-rotations and/or 3D-reflections; so exercise 300 is used instead of exercise 240 to generate the options of the problem. It turns out that there are 688 options, involving 34 items that we can call `1`, `2`, . . . , `7`, `000`, `001`, . . . , `222`. For example, the first option

$$‘\texttt{1}\quad\texttt{000}\quad\texttt{001}\quad\texttt{010}’ \qquad (40)$$

characterizes one of the 144 potential ways to place the "bent" piece 1.

Algorithm D needs just 407 megamems to find all 11,520 solutions to this problem. Furthermore, we can save most of that time by taking advantage of symmetry: Every solution can be rotated into a unique "canonical" solution in which the "ell" piece 2 has not been rotated; hence we can restrict that piece to only six placements, namely (`000`, `010`, `020`, `100`), (`001`, `011`, `021`, `101`), . . . , (`102`, `112`, `122`, `202`) — all shifts of each other. This restriction removes $138 = \frac{23}{24}\cdot 144$ options, and the algorithm now finds the 480 canonical solutions in just 20 megamems. (These canonical solutions form 240 mirror-image pairs.)

Factoring an exact cover problem. In fact, we can simplify the Soma cube problem much further, so that all of its solutions can actually be found by hand in a reasonable time, by *factoring* the problem in a clever way.

Let's observe first that any solution to an exact cover problem automatically solves infinitely many *other* problems. Going back to our original formulation in terms of an $m \times n$ matrix $A = (a_{ij})$, the task is to find all sets of rows whose sum is 1 in every column, namely to find all binary vectors $x_1 \ldots x_m$ such that $\sum_{i=1}^{m} x_i a_{ij} = 1$ for $1 \le j \le n$. Therefore if we set $b_i = \alpha_1 a_{i1} + \cdots + \alpha_n a_{in}$ for $1 \le i \le m$, where $(\alpha_1, \ldots, \alpha_n)$ is any n-tuple of coefficients, the vectors $x_1 \ldots x_m$ will also satisfy $\sum_{i=1}^{m} x_i b_i = \alpha_1 + \cdots + \alpha_n$. By choosing the α's intelligently we may be able to learn a lot about the possibilities for $x_1 \ldots x_m$.

For example, consider again the 6×7 matrix A in (5), and let $\alpha_1 = \cdots = \alpha_7 = 1$. The sum of the entries in each row of A is either 2 or 3; thus we're supposed to cover 7 things, by burying either 2 or 3 at a time. Without knowing anything more about the detailed structure of A, we can conclude immediately that there's only one way to obtain a total of 7, namely by selecting $2 + 2 + 3$! Furthermore, only rows 1 and 5 have 2 as their sum; we *must* choose them.

chessboard
straight trominoes
monomino
Golomb
coloring arguments
Slothouber
Graatsma
square tetracubes

Now here's a more interesting challenge: "Cover the 64 cells of a chessboard with 21 straight trominoes and one monomino." This problem corresponds to a big matrix that has $96+64$ rows and $1+64$ columns,

$$\begin{array}{c}
\begin{array}{cccccccccccccccccccccccccccc}
\mathtt{M} & \mathtt{00} & \mathtt{01} & \mathtt{02} & \mathtt{03} & \mathtt{04} & \mathtt{05} & \mathtt{06} & \mathtt{07} & \mathtt{10} & \mathtt{11} & \mathtt{12} & \mathtt{13} & \mathtt{14} & \mathtt{15} & \mathtt{16} & \mathtt{17} & \mathtt{20} & \mathtt{21} & \mathtt{22} & \mathtt{23} & \mathtt{24} & & \mathtt{74} & \mathtt{75} & \mathtt{76} & \mathtt{77}
\end{array} \\
\begin{pmatrix}
0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\ldots&0&0&0&0\\
0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&\ldots&0&0&0&0\\
0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\ldots&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&\ldots&0&0&0&0\\
&&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&&&\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\ldots&0&1&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\ldots&0&0&0&0\\
1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\ldots&0&0&0&0\\
&&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&&&\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\ldots&0&0&0&1
\end{pmatrix}
\end{array}, \qquad (41)$$

where the first 96 rows specify all possible ways to place a tromino and the other 64 rows specify the possibilities for the monomino. Column ij represents cell (i,j); column '$\mathtt{M}$' means "monomino."

The three cells (i,j) covered by a straight tromino always lead to distinct values of $(i-j) \bmod 3$. Therefore, if we add up the 22 columns of (41) for which $(i-j) \bmod 3 = 0$, we get 1 in each of the first 96 rows, and 0 or 1 in the other 64 rows. We're supposed to get a total of 22 in the chosen rows; hence the monomino has to go into a cell (i,j) with $i \equiv j$ (modulo 3).

A similar argument, using $i+j$ instead of $i-j$, shows that the monomino must also go into a cell with $i+j \equiv 1$ (modulo 3). Therefore $i \equiv j \equiv 2$. *We've proved that there are only four possibilities for* (i,j), namely $(2,2)$, $(2,5)$, $(5,2)$, $(5,5)$. [Golomb made this observation in his 1954 paper that introduced polyominoes, after "coloring" the cells of a chessboard with three colors. The general notion of factoring includes all such coloring arguments as special cases.]

Our proof that (38) is an impossible pentomino placement can also be regarded as an instance of factorization. The residual problem, if (38) is chosen, has a total of either 0 or 5 in the first 27 columns of each remaining row of the associated matrix. Therefore we can't achieve a total of 27 from those rows.

Consider now a three-dimensional problem [J. Slothouber and W. Graatsma, *Cubics* (1970)]: Can six square tetracubes (that is, six $2\times2\times1$ cuboids) be packed into a $3\times3\times3$ box? This is the problem of choosing six rows of the 36×27 matrix

$$\begin{array}{c}
\begin{array}{ccccccccccccccccccccccccccc}
\mathtt{000}&\mathtt{001}&\mathtt{002}&\mathtt{010}&\mathtt{011}&\mathtt{012}&\mathtt{020}&\mathtt{021}&\mathtt{022}&\mathtt{100}&\mathtt{101}&\mathtt{102}&\mathtt{110}&\mathtt{111}&\mathtt{112}&\mathtt{120}&\mathtt{121}&\mathtt{122}&\mathtt{200}&\mathtt{201}&\mathtt{202}&\mathtt{210}&\mathtt{211}&\mathtt{212}&\mathtt{220}&\mathtt{221}&\mathtt{222}
\end{array} \\
\begin{pmatrix}
1&1&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&1&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&1&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&1&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&1&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
&&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&&\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&1&1
\end{pmatrix}
\end{array}, \qquad (42)$$

in such a way that all of the column sums are ≤ 1.

Soma cube
checkerboard coloring

The 27 cubies (i,j,k) of a $3 \times 3 \times 3$ cube fall into four classes, depending on how many of its coordinates have the middle value 1:

$$\begin{array}{ll} \text{A vertex cubie has no 1s.} & \bigl(\binom{3}{0}2^3 = 8 \text{ cases.}\bigr) \\ \text{An edge cubie has one 1.} & \bigl(\binom{3}{1}2^2 = 12 \text{ cases.}\bigr) \\ \text{A face cubie has two 1s.} & \bigl(\binom{3}{2}2^1 = 6 \text{ cases.}\bigr) \\ \text{A central cubie has three 1s.} & \bigl(\binom{3}{3}2^0 = 1 \text{ case.}\bigr) \end{array} \qquad (43)$$

Every symmetry of the cube preserves these classes.

Imagine placing four new columns v, e, f, c at the right of (42), representing the number of vertex, edge, face, and central cubies of a placement. Then 24 of the rows will have $(v,e,f,c) = (1,2,1,0)$, and the other 12 rows will have $(v,e,f,c) = (0,1,2,1)$. If we choose a rows of the first kind and b rows of the second kind, this factorization tells us that we must have

$$a \ge 0,\ \ b \ge 0,\ \ a+b=6,\ \ a \le 8,\ \ 2a+b \le 12,\ \ a+2b \le 6,\ \ b \le 1. \qquad (44)$$

That's more than enough to prove that $b = 0$ and $a = 6$, and thus to find the essentially unique solution.

(We could paraphrase this argument as follows, making it more impressive by concealing the low-level algebra that inspired it: "Each tetracube occupies at least one face cubie. So each of them must be placed on a different face.")

With these examples in mind, we're ready now to apply factorization to the Soma cube. The possible (v,e,f,c) values for pieces 1 through 7 in (39) are:

$$\begin{array}{ll} \text{Piece 1:} & (0,1,1,1),\ (0,0,2,1),\ (0,1,2,0),\ (0,2,1,0),\ (1,1,1,0),\ (1,2,0,0). \\ \text{Piece 2:} & (0,1,2,1),\ (0,2,2,0),\ (1,2,1,0),\ (2,2,0,0). \\ \text{Piece 3:} & (0,0,3,1),\ (0,2,1,1),\ (0,3,1,0),\ (2,1,1,0). \\ \text{Piece 4:} & (0,1,2,1),\ (1,2,1,0). \\ \text{Piece 5:} & (0,1,2,1),\ (0,2,2,0),\ (1,1,1,1),\ (1,2,1,0). \\ \text{Piece 6:} & (0,1,2,1),\ (0,2,2,0),\ (1,1,1,1),\ (1,2,1,0). \\ \text{Piece 7:} & (0,2,1,1),\ (0,3,0,1),\ (1,1,2,0),\ (1,3,0,0). \end{array} \qquad (45)$$

(This is actually much more information than we need, but it doesn't hurt.)

Looking only at the totals for v, we see that we must have

$$(0 \text{ or } 1) + (0,\ 1, \text{ or } 2) + (0 \text{ or } 2) + (0 \text{ or } 1) + (0 \text{ or } 1) + (0 \text{ or } 1) + (0 \text{ or } 1) \ = \ 8;$$

and the *only* way to achieve this is via

$$(0 \text{ or } 1) + (1 \text{ or } 2) + 2 + (0 \text{ or } 1) + (0 \text{ or } 1) + (0 \text{ or } 1) + (0 \text{ or } 1) \ = \ 8,$$

thus eliminating several options for pieces 2 and 3. More precisely, *piece 2 must touch at least one vertex; piece 3 must be placed along an edge.*

Looking next at the totals for $v+f$, which are the "black" cubies if we color them alternately black and white with black in the corners, we must also have

$$(1 \text{ or } 2) + 2 + 3 + 2 + 2 + 2 + (1 \text{ or } 3) \ = \ 14;$$

and the only way to achieve this is with two from piece 1 and one from piece 7: *Piece 1 must occupy two black cubies, and piece 7 must occupy just one.*

We have therefore eliminated 200 of the 688 options that begin with (40). And we also know that exactly five of the pieces 1, 2, 4, 5, 6, 7 occupy as many of the corner vertices as they individually can. This extra information can be encoded by introducing 13 new primary items

`*, 1+, 1-, 2+, 2-, 4+, 4-, 5+, 5-, 6+, 6-, 7+, 7-` (46)

and six new options

```
'* 1+ 2- 4- 5- 6- 7-'
'* 1- 2+ 4- 5- 6- 7-'
'* 1- 2- 4+ 5- 6- 7-'
'* 1- 2- 4- 5+ 6- 7-'      (47)
'* 1- 2- 4- 5- 6+ 7-'
'* 1- 2- 4- 5- 6- 7+'
```

and by appending *p*+ or *p*− to each of piece *p*'s options that do or don't touch the most corners. For example, this new set of $6 + 488$ options for the Soma cube problem includes the following typical ways to place various pieces:

```
'1 000 001 011 1+'
'1 001 011 101 1-'
'2 000 001 002 010 2+'
'2 000 001 011 021 2-'
'3 000 001 002 011'
'4 000 001 011 012 4+'
'4 000 011 111 121 4-'
'5 000 001 010 110 5+'
'5 001 010 011 101 5-'
'6 000 001 010 101 6+'
'6 001 010 011 110 6-'
'7 000 001 010 100 7+'
'7 001 010 011 111 7-'
```

As before, Algorithm D finds 11,520 solutions; but now it needs only 108 megamems to do so. Each of the new options is used in at least 21 of the solutions, hence we've removed all of the "fat" in the original set. [This instructive analysis of Soma is due to M. J. T. Guy, R. K. Guy, and J. H. Conway in 1961. See Berlekamp, Conway, and Guy, *Winning Ways*, second edition (2004), 845–847.]

To reduce the number of solutions, using symmetry, we can force piece 3 to occupy {000, 001, 002, 011} (thus saving a factor of 24), and remove all options for piece 7 that use a cell ijk with $k = 2$ (saving an additional factor of 2). From the remaining 455 options, Algorithm D needs just 2 megamems to generate all 240 of the essentially distinct solutions.

The seven Soma pieces are amazingly versatile, and so are the other polycubes of small sizes. Exercises 300–324 explore some of their remarkable properties, together with historical references.

Guy, Michael
Guy
Conway
Berlekamp
Conway
Guy
symmetry

Color-controlled covering. *Take a break!* Before reading any further, please spend a minute or two solving the "word search" puzzle in Fig. 71. Comparatively mindless puzzles like this one provide a low-stress way to sharpen your word-recognition skills. It can be solved easily — for instance, by making eight passes over the array — and the solution appears in Fig. 72.

color-controlled–
word search
color codes

Fig. 71. Find the mathematicians*:

Put ovals around the following names where they appear in the 15 × 15 array shown here, reading either forward or backward or upward or downward, or diagonally in any direction. After you've finished, the leftover letters will form a hidden message. (The solution appears on the next page.)

ABEL	HENSEL	MELLIN
BERTRAND	HERMITE	MINKOWSKI
BOREL	HILBERT	NETTO
CANTOR	HURWITZ	PERRON
CATALAN	JENSEN	RUNGE
FROBENIUS	KIRCHHOFF	STERN
GLAISHER	KNOPP	STIELTJES
GRAM	LANDAU	SYLVESTER
HADAMARD	MARKOFF	WEIERSTRASS

O	T	H	E	S	C	A	T	A	L	A	N	D	A	U
T	S	E	A	P	U	S	T	H	O	R	S	R	O	F
T	L	S	E	E	A	Y	R	R	L	Y	H	A	P	A
E	P	E	A	R	E	L	R	G	O	U	E	M	S	I
N	N	A	R	R	C	V	L	T	R	T	A	A	M	A
I	T	H	U	O	T	E	K	W	I	A	N	D	E	M
L	A	N	T	N	B	S	I	M	I	C	M	A	A	W
L	G	D	N	A	R	T	R	E	B	L	I	H	C	E
E	R	E	C	I	Z	E	C	E	P	T	N	E	D	Y
M	E	A	R	S	H	R	H	L	I	P	K	A	T	H
E	J	E	N	S	E	N	H	R	I	E	O	N	E	T
H	S	U	I	N	E	B	O	R	F	E	W	N	A	R
T	M	A	R	K	O	F	F	O	F	C	S	O	K	M
P	L	U	T	E	R	P	F	R	O	E	K	G	R	A
G	M	M	I	N	S	E	J	T	L	E	I	T	S	G

Our goal in this section is not to discuss how to *solve* such puzzles; instead, we shall consider how to *create* them. It's by no means easy to pack those 27 names into the box in such a way that their 184 characters occupy only 135 cells, with eight directions well mixed. How can that be done with reasonable efficiency?

For this purpose we shall extend the idea of exact covering by introducing "color codes." ...

Who knows what I might eventually decide to say next? Please stay tuned.

* * *

* The journal *Acta Mathematica* celebrated its 21st birthday by publishing a special *Table Générale des Tomes 1–35*, edited by Marcel Riesz (Uppsala: 1913), 179 pp. It contained a complete list of all papers published so far in that journal, together with portraits and brief biographies of all the authors. The 27 mathematicians mentioned in Fig. 71 are those who were subsequently mentioned in Volumes 1, 2, or 3 of *The Art of Computer Programming* — except for people like MITTAG-LEFFLER or POINCARÉ, whose names contain special characters.

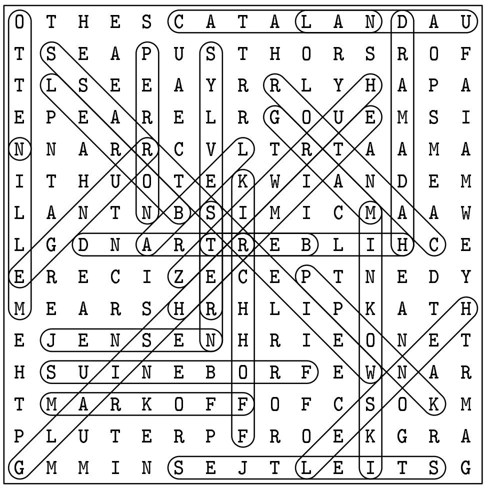
Fig. 72. Solution to the puzzle of the hidden mathematicians (Fig. 71). Notice that the central letter `R` actually participates in six different names:

```
BERTRAND
GLAISHER
HERMITE
HILBERT
KIRCHHOFF
WEIERSTRASS
```

The `T` to its left participates in five.

Here's what the leftover letters say:

These authors of early papers in *Acta Mathematica* were cited years later in *The Art of Computer Programming.*



Hitotumatu
Noshita
N queens problem
Rohl
singly linked lists
Knuth
Hoare
DLX

* * *

Historical notes. The basic idea of (2) was introduced by H. Hitotumatu and K. Noshita [*Information Processing Letters* **8** (1979), 174–175], who applied it to the N queens problem. Algorithm 7.2.1.2X, which was published by J. S. Rohl in 1983, can be regarded as a simplified version of dancing links, for cases when singly linked lists suffice. (Indeed, as Rohl observed, the N queens problem is such a case.) Its extension to exact cover problems in general, as in Algorithm D above, was the subject of D. E. Knuth's tribute to C. A. R. Hoare in *Millennial Perspectives in Computer Science* (2000), 187–214, where numerous examples were given. His original implementation, called DLX, used a more complex data structure than (10), involving nodes with four-way links.

EXERCISES — First Set

▶ **1.** [*M25*] A doubly linked list of n elements, with a list head at 0, begins with `LLINK`$(k) = k-1$ and `RLINK`$(k-1) = k$ for $1 \le k \le n$, `LLINK(0)` $= n$, and `RLINK`$(n) = 0$, as in (3). But after we use operation (1) to delete elements $a_1, a_2, \ldots, a_n$, where $a_1a_2\ldots a_n$ is a permutation of $\{1, 2, \ldots, n\}$, the list will be empty and the links will be entangled as in (4).

a) Show that the final settings of `LLINK` and `RLINK` can be described in terms of the binary search tree that is obtained when the keys $a_n, \ldots, a_2, a_1$ (in reverse order) are inserted by Algorithm 6.2.2T into an initially empty tree.

b) Say that permutations $a_1a_2\ldots a_n$ and $b_1b_2\ldots b_n$ are equivalent if they both yield the same `LLINK` and `RLINK` values after deletion. How many distinct equivalence classes arise, for a given value of n?

c) How many of those equivalence classes contain just one permutation?

2. [*M30*] Continuing exercise 1, we know that the original list will be restored if we use (2) to undelete the elements $a_n, \ldots, a_2, a_1$, reversing the order of deletion.

a) Prove that it's restored *also* if we use the unreversed order $a_1, a_2, \ldots, a_n$ (!).

b) Is the original list restored if we undelete the elements in *any* order whatsoever?

3. [*20*] An $m \times n$ matrix that's supposed to be exactly covered can be regarded as a set of n simultaneous equations in m unknowns. For example, (5) is equivalent to

$$x_2 + x_4 = x_3 + x_5 = x_1 + x_3 = x_2 + x_4 + x_6 = x_1 + x_6 = x_3 + x_4 = x_2 + x_5 + x_6 = 1,$$

where each $x_k =$ [choose row k] is either 0 or 1.

a) What is the general solution to those seven equations?

b) Why is this approach to exact cover problems almost never useful in practice?

4. [*M20*] Given a graph G, construct a matrix with one row for each vertex v and one column for each edge e, putting the value [e touches v] into column e of row v. What do the exact covers of this matrix represent?

5. [*18*] Among the many combinatorial problems that can be formulated in terms of 0–1 matrices, some of the most important deal with *families of sets*: The columns of the matrix represent elements of a given universe, and the rows represent subsets of that universe. The exact cover problem is then to partition the universe into such subsets. Equivalently, we can use the terminology of hypergraphs, speaking of hyperedges (rows) that consist of vertices (columns); then the exact cover problem is to find a perfect matching, also called a perfect packing, namely a set of nonoverlapping hyperedges that hit every vertex.

Such problems generally have *duals*, which arise when we transpose the rows and columns of the input matrix. What is the dual of the exact cover problem, in hypergraph terminology?

6. [*15*] If an exact cover problem has N items and M options, and if the total length of all options is L, how many nodes are in the data structures used by Algorithm D?

7. [*16*] Why is `TOP(23)` $= -4$ in Table 1? Why is `DLINK(23)` $= 25$?

8. [*22*] Design an algorithm to set up the initial memory contents of an exact cover problem, as needed by Algorithm D and illustrated in Table 1. The input to your algorithm should consist of a sequence of lines with the following format:

- The very first line lists the names of all items.
- Each remaining line specifies the items of a particular option, one option per line.

doubly linked list
list head
LLINK
RLINK
delete
permutation
binary search tree
undelete
linear equations
0–1 matrices
families of sets
hypergraphs
hyperedges
vertices
perfect matching
matching, perfect
packing
duals

9. [*18*] Explain how to branch in step D3 on an item i for which `LEN(`i`)` is minimum. If several items have that minimum length, i itself should also be minimum. (This choice is often called the "minimum remaining values" (MRV) heuristic.)

▶ **10.** [*19*] Play through Algorithm D by hand, using exercise 9 in step D3 and the input in Table 1, until first reaching step D7. What are the contents of memory at that time?

▶ **11.** [*21*] Design an algorithm that prints the option associated with a given node x, cyclically ordering the option so that `TOP(`x`)` is its first item. Also print the position of that option in the vertical list for that item. (For example, if $x = 21$ in Table 1, your algorithm should print '$d\ f\ a$' and state that it's option 2 of 3 in the list for item d.)

12. [*16*] When Algorithm D finds a solution in step D2, how can we use the values of $x_0x_1 \ldots x_{l-1}$ to figure out what that solution is?

▶ **13.** [*M30*] The running time of Algorithm D is largely determined by the number of *updates* that it makes to its data structures, namely the number of times that it removes (and later restores) a node or record from a doubly linked list. This is the number of times hide(p) sets `LEN(`x`)` $\leftarrow$ `LEN(`x`)` $- 1$, plus the number of times cover(i) is called.

How many updates does Algorithm D make when it is applied to the "extreme exact cover problem," which has n items and $2^n - 1$ distinct options? (In this case *all* partitions of the set $\{1, \ldots, n\}$ are solutions; see Section 7.2.1.5.)

14. [*20*] The options in (16) give us every solution to the Langford pair problem twice, because the left-right reversal of any solution is also a solution. Show that, if a few of those options are removed, we'll get only half as many solutions; the others will be the reversals of the solutions found.

15. [*16*] What are the solutions to the four queens problem, as formulated in (23) and (24)? What branches are taken at the top four levels of Algorithm D's search tree?

16. [*16*] Repeat exercise 15, but consider a_j and b_j to be secondary items and omit the slack options (24). Consider the primary items in order r_3, c_3, r_2, c_2, r_4, c_4, r_1, c_1.

17. [*10*] What are the solutions to (6) if items e, f, and g are *secondary*?

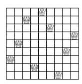
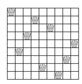
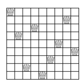
▶ **18.** [*21*] Modify Algorithm D so that it doesn't require the presence of any primary items in the options. A valid solution should not contain any purely secondary options; but it must intersect every such option. (For example, if only items a and b of (6) were primary, the only valid solution would be to choose options '$a\ d\ g$' and '$b\ c\ f$'.)

▶ **19.** [*25*] Generalize (26) to a pairwise ordering of options $(\alpha_0, \ldots, \alpha_{m-1}; \beta_0, \ldots, \beta_{m-1})$ that uses at most $\lceil \lg m \rceil$ of the secondary items $y_1, \ldots, y_{m-1}$ in each option. *Hint:* Think of binary notation, and use y_j at most $2^{\rho j}$ times within each of the α's and β's.

20. [*22*] Extend exercise 19 to k-wise ordering of km options α_j^i, for $1 \le i \le k$ and $0 \le j \le m$. The solutions should be $(\alpha_{j_1}^1, \ldots, \alpha_{j_k}^k)$ with $0 \le j_1 \le \cdots \le j_k < m$. Again there should be at most $\lceil \lg m \rceil$ secondary items in each option.

▶ **21.** [*28*] Most of the solutions to the n queens problem are unsymmetrical, hence they lead to seven other solutions when rotated and/or reflected. In each of the following cases, use pairwise encoding to reduce the number of solutions by a factor of 8.

a) No queen is in either diagonal, and n is odd.
b) Only one of the two diagonals contains a queen.
c) There are two queens in the two diagonals.

(a)  (b)  (c) 

22. [*28*] Use pairwise encoding to reduce the number of solutions by *nearly* a factor of 8 in the remaining cases not covered by exercise 21:

minimum remaining values
MRV
updates
extreme exact cover problem
partitions of the set
set partitions
Langford pair
queens
secondary items
secondary
pairwise ordering
binary notation
ruler function ρ
k-wise ordering
unsymmetrical
symmetry breaking
canonical solutions
diagonal

a) No queen is in either diagonal, and n is even.
b) A queen is in the center of the board, and n is odd.

23. [*20*] With Algorithm D, find all solutions to the n queens problem that are unchanged when they're rotated by (a) $180°$; (b) $90°$.

24. [*20*] By setting up an exact cover problem and solving it with Algorithm D, show that the queen graph Q_8 (exercise 7.1.4–241) cannot be colored with eight colors.

25. [*21*] In how many ways can Q_8 be colored in a "balanced" fashion, using eight queens of color 0 and seven each of colors 1 to 8?

26. [*22*] Introduce secondary items cleverly into the options (16), so that only *planar* solutions to Langford's problem are obtained. (See exercise 7–8.)

27. [*M22*] For what integers c_0, t_0, c_1, t_1, $\ldots$, c_l, t_l with $1 \le c_j \le t_j$ does the text's formula (27) for estimated completion ratio give the value (a) $1/2$? (b) $1/3$?

▸ **28.** [*26*] Let T be any tree. Construct the 0–1 matrix of an unsolvable exact cover problem for which T is the backtrack tree traversed by Algorithm D with the MRV heuristic. (A *unique* item should have the minimum `LEN` value whenever step D3 is encountered.) Illustrate your construction when $T =$.

29. [*25*] Continuing exercise 28, let T be a tree in which certain leaves have been distinguished from the others and designated as "solutions."
a) Show that some trees of that kind never match the behavior of Algorithm D.
b) Characterize all such trees that *do* arise, having solutions where indicated.

31. [*M21*] The solution to an exact cover problem with M options can be regarded as a binary vector $x = x_1 \ldots x_M$, with $x_k = [\text{choose option } k]$. The *distance* between two solutions x and x' can then be defined as the Hamming distance $d(x,x') = \nu(x \oplus x')$, the number of places where x and x' differ. The *diversity* of the problem is the minimum distance between two of its solutions. (If there's at most one solution, the diversity is ∞.)
a) Is it possible to have diversity 1?
b) Is it possible to have diversity 2?
c) Is it possible to have diversity 3?
d) Prove that the distance between solutions of a *uniform* exact cover problem — that is, a problem having the same number of items in each option — is always even.
e) Most of the exact cover problems that arise in applications are at least *quasi-uniform*, in the sense that they have a nonempty subset of primary items such that the problem is uniform when restricted to only those items. (For example, every polyomino or polycube packing problem is quasi-uniform, because every option specifies exactly one piece name.) Can such problems have odd distances?

33. [*M16*] Given an exact cover problem, specified by a 0–1 matrix A, construct an exact cover problem A' that has exactly one more solution than A does. [Consequently it is NP-hard to determine whether an exact cover problem with at least one solution has more than one solution.] Assume that A contains no all-zero rows.

34. [*M25*] Given an exact cover problem A as in exercise 33, construct an exact cover problem A' such that (i) A' has at most three 1s in every column; (ii) A' and A have exactly the same number of solutions.

35. [*M21*] Continuing exercise 34, construct A' having *exactly* three 1s per column.

▸ **36.** [*30*] Given an $m \times n$ exact cover problem A with exactly three 1s per column (see exercise 35), construct a generalized "instant insanity" problem with $N = O(n)$ cubes and N colors that is solvable if and only if A is solvable. (See 7.2.2–(36).)

diagonal
central symmetry
90-deg rot symmetry
queen graph
colored
secondary items
Langford's problem
completion ratio
0–1 matrix
backtrack tree
MRV
distance
Hamming distance
diversity
uniform
quasi-uniform
polyomino
polycube
0–1
NP-hard
unique solution
instant insanity

▶ **50.** [*21*] If we merely want to count the number of solutions to an exact cover problem, without actually constructing them, a completely different approach based on bitwise manipulation instead of list processing is sometimes useful.

The following naïve algorithm illustrates the idea: We're given an $m \times n$ matrix of 0s and 1s, represented as n-bit vectors $r_1, \ldots, r_m$. The algorithm works with a (potentially huge) database of pairs (s_j, c_j), where s_j is an n-bit number representing a set of items, and c_j is a positive integer representing the number of ways to cover that set exactly. Let p be the n-bit mask that represents the primary items.

N1. [Initialize.] Set $N \leftarrow 1$, $s_1 \leftarrow 0$, $c_1 \leftarrow 1$, $k \leftarrow 1$.

N2. [Done?] If $k > m$, terminate; the answer is $\sum_{j=1}^{N} c_j\,[s_j \,\&\, p = p]$.

N3. [Append r_k where possible.] Set $t \leftarrow r_k$. For $N \geq j \geq 1$, if $s_j \,\&\, t = 0$, insert $(s_j + t, c_j)$ into the database (see below).

N4. [Loop on k.] Set $k \leftarrow k + 1$ and return to N2. ▌

To insert (s, c) there are two cases: If $s = s_i$ for some (s_i, c_i) already present, we simply set $c_i \leftarrow c_i + c$. Otherwise we set $N \leftarrow N + 1$, $s_N \leftarrow s$, $c_N \leftarrow c$.

Show that this algorithm can be significantly improved by using the following trick: Set $u_k \leftarrow r_k \,\&\, \bar{f}_k$, where $f_k = r_{k+1} \mid \cdots \mid r_m$ is the bitwise OR of all future rows. If $u_k \neq 0$, we can remove any entry from the database for which s_j does not contain $u_k \& p$. We can also exploit the nonprimary items of u_k to compress the database further.

51. [*25*] Implement the improved algorithm of the previous exercise, and compare its running time to that of Algorithm D when applied to the n queens problem.

52. [*M21*] Explain how the method of exercise 50 could be extended to give representations of all solutions, instead of simply counting them.

59. [*M20*] Give formulas for the entries a_{ij}, b_{ij}, c_{ij} of the sudoku squares in (28).

60. [*M04*] Could the clues of a sudoku puzzle be the first 33 digits of π? (See (29a).)

61. [*10*] List the sequence of naked single moves by which Algorithm D cruises to a solution of (29a). (If several such p_{ij} are possible, choose the smallest ij at each step.)

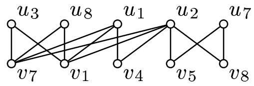
62. [*19*] List all of the *hidden* single sudoku moves that are present in chart (31).

63. [*19*] What hidden singles are present in (32), after '3' is placed in cell $(2, 3)$?

▶ **64.** [*24*] Chart (33) essentially plots rows versus columns. Show that the same data could be plotted as either (a) rows versus values; or (b) values versus columns.

▶ **65.** [*24*] Any solution to an exact cover problem will also solve the "relaxed" subproblems that are obtained by removing some of the items. For example, we might relax a sudoku problem (30) by removing all items c_{jk} and b_{xk}, as well as r_{ik} with $i \neq i_0$. Then we're left with a subproblem in which every option contains just two items, '$p_{i_0 j}\ r_{i_0 k}$', for certain pairs (j, k). In other words, we're left with a 2D matching problem.

Consider the bipartite graph with u_j — v_k whenever a sudoku option contains '$p_{i_0 j}\ r_{i_0 k}$'. For example, the graph for $i_0 = 4$ in (33) is illustrated below. A perfect matching of this graph must take u_3 and u_8 to either v_7 or v_1, hence the edges from other u's to those v's can be deleted; that's called a "naked pair" in row i_0. Dually, v_5 and v_8 must be matched to either u_2 or u_7, hence the edges from other v's to those u's can be deleted; that's called a "hidden pair" in row i_0.



In general, q of the u's form a *naked q-tuple* if their neighbors include only q of the v's; and q of the v's form a *hidden q-tuple* if their neighbors include only q of the u's.

exact cover problem
bitwise manipulation
breadth-first
0s and 1s
primary items
bitwise AND
bitwise OR
nonprimary items
n queens problem
sudoku squares
sudoku puzzle
naked single
hidden single moves
relaxed
2D matching problem
bipartite graph
naked pair
hidden pair

a) These definitions have been given for rows. Show that naked and hidden q-tuples can be analogously for (i) columns, (ii) boxes.

b) Prove that if the bipartite graph has r vertices in each part, it has a hidden q-tuple if and only if it has a naked $(r-q)$-tuple.

c) Find all the naked and hidden q-tuples of (33). What options do they rule out?

d) Consider deleting items p_{ij} and b_{xk}, as well as all r_{ik} and c_{jk} for $k \neq k_0$. Does this lead to further reductions of (33)?

66. [*20*] How many uniquely solvable 17-clue puzzles contain the 16 clues of (29c)?

67. [*22*] In how many ways can (29c) be completed so that every row, every column, and every box contains a permutation of the multiset $\{1, 2, 3, 4, 5, 6, 7, 7, 9\}$?

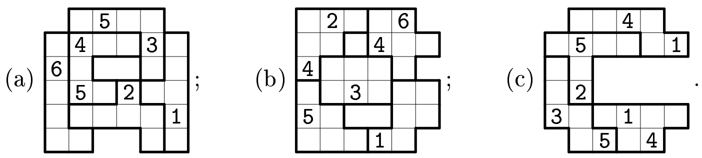
▸ **68.** [*35*] (*Minimal clues.*) Puzzle (29a) contains more clues than necessary to make the sudoku solution unique. (For example, the final '95' could be omitted.) Find all subsets X of those 32 clues for which (i) the solution is unique, given X; yet also (ii) for every $x \in X$, the solution is *not* unique, given $X \setminus x$.

69. [*34*] (G. McGuire.) Prove that at least 18 clues are necessary, in any sudoku puzzle whose unique answer is (28a). Also find 18 clues that suffice. *Hint:* At least two of the nine appearances of $\{1, 4, 7\}$ in the top three rows must be among the clues.

Similarly, find a smallest-possible set of clues whose unique answer is (28b).

70. [*20*] Solve the jigsaw sudokus in (34). How large is Algorithm D's search tree?

71. [*20*] (*The Puzzlium Sudoku ABC.*) Complete these hexomino-shaped boxes:

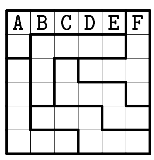


72. [*21*] Turn Behrens's 5×5 gerechte design (35a) into a puzzle, by erasing all but five of its 25 entries.

▸ **73.** [*34*] For $n \leq 7$, generate all of the ways in which an $n \times n$ square can be packed with n nonstraight n-ominoes. (These are the possible arrangements of boxes in a square jigsaw sudoku.) How many of them are symmetric? *Hint:* See exercise 7.2.2–101.

74. [*29*] In how many different ways can Behrens's 9×9 array (35c) be regarded as a gerechte latin square? (In other words, how many decompositions of that square into nine boxes of size 9 have a complete "rainbow" $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ in each box? None of the boxes should simply be an entire row or an entire column.)

75. [*23*] (*Clueless jigsaw sudoku.*) A jigsaw sudoku puzzle can be called "clueless" if its solution is uniquely determined by the entries in a single row or column, because such clues merely assign names to the n individual symbols that appear. For example, the first such puzzle to be published, discovered in 2000 by Oriel Maxime, is shown here.

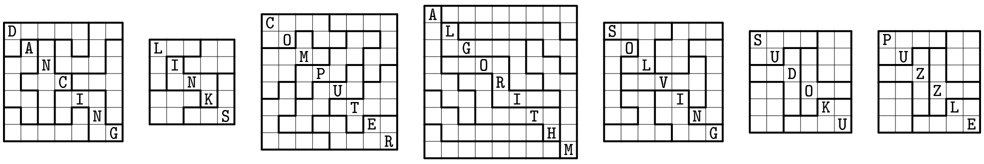


a) Find all clueless sudoku jigsaw puzzles of order $n \leq 6$.

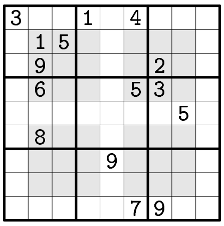
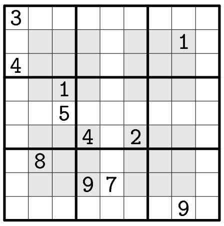
b) Prove that such puzzles exist of all orders $n \geq 4$.

multiset
McGuire
jigsaw sudokus
Puzzlium Sudoku ABC
hexomino
alphabet
Behrens
gerechte design
n-ominoes
boxes
rainbow
Clueless jigsaw sudoku
Maxime

76. [*24*] Find the unique solutions to the following examples of jigsaw sudoku:



▸ **78.** [*22*] *Hypersudoku* extends normal sudoku by adding four more (shaded) boxes in which a complete "rainbow" $\{1,2,3,4,5,6,7,8,9\}$ is required to appear:

(i)  (ii) 

(Such puzzles, introduced by P. Ritmeester in 2005, are featured by many newspapers.)

a) Show that a hypersudoku solution actually has 18 rainbow boxes, not only 13.
b) Use that observation to solve hypersudoku puzzles efficiently by extending (30).
c) How much does that observation help when solving (i) and (ii)?
d) True or false: A hypersudoku solution remains a hypersudoku solution if the four 4×4 blocks that touch its four corners are simultaneously rotated 180°, while also flipping the middle half-rows and middle half-columns (keeping the center fixed).

▸ **97.** [*M24*] A *grope* is a set G together with a binary operation $\circ$, in which the identity $x \circ (y \circ x) = y$ is satisfied for all $x \in G$ and $y \in G$.

a) Prove that the identity $(x \circ y) \circ x = y$ also holds, in every grope.
b) Which of the following "multiplication tables" define a grope on $\{0,1,2,3\}$?

$$\begin{matrix}0123\\1032\\2301\\3210\end{matrix};\quad \begin{matrix}0321\\3210\\2103\\1032\end{matrix};\quad \begin{matrix}0132\\1023\\3210\\2301\end{matrix};\quad \begin{matrix}0231\\3102\\1320\\2013\end{matrix};\quad \begin{matrix}0312\\2130\\3021\\1203\end{matrix}.$$

(In the first example, $x \circ y = x \oplus y$; in the second, $x \circ y = (-x-y) \bmod 4$. The last two have $x \circ y = x \oplus f(x \oplus y)$ for certain functions f.)

c) For all n, construct a grope whose elements are $\{0,1,\ldots,n-1\}$.
d) Consider the exact cover problem that has n^2 items xy for $0 \le x, y < n$ and the following $n + (n^3 - n)/3$ options:
   i) 'xx', for $0 \le x < n$;
   ii) '$xx\ xy\ yx$', for $0 \le x < y < n$;
   iii) '$xy\ yz\ zx$', for $0 \le x < y, z < n$.

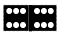
   Show that its solutions are in one-to-one correspondence with the multiplication tables of gropes on the elements $\{0,1,\ldots,n-1\}$.
e) Element x of a grope is *idempotent* if $x \circ x = x$. If k elements are idempotent and $n-k$ are not, prove that $k \equiv n^2$ (modulo 3).

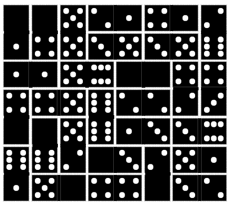
98. [*21*] Modify the exact cover problem of exercise 97(d) in order to find the multiplication tables of (a) all idempotent gropes — gropes such that $x \circ x = x$ for all x;

jigsaw sudoku
sudoku
Hypersudoku
NRC Sudoku, see hypersudoku
rainbow
Ritmeester
pi as random data
grope
binary operation
multiplication tables
idempotent

(b) all commutative gropes — gropes such that $x \circ y = y \circ x$ for all x and y; (c) all gropes with an identity element — gropes such that $x \circ 0 = 0 \circ x = x$ for all x.

▶ **99.** [*HM00*] this is a temporary exercise (for dummies)

142. [*21*] *Dominosa* is a solitaire game in which you “shuffle” the 28 pieces , , ...,  of double-six dominoes and place them at random into a 7×8 frame. Then you write down the number of spots in each cell, put the dominoes away, and try to reconstruct their positions based only on that 7×8 array of numbers. For example,

 yields the array
$$\begin{pmatrix} 0&0&5&2&1&4&1&2\\ 1&4&5&3&5&3&5&6\\ 1&1&5&6&0&0&4&4\\ 4&4&5&6&2&2&2&3\\ 0&0&5&6&1&3&3&6\\ 6&6&2&0&3&2&5&1\\ 1&5&0&4&4&0&3&2 \end{pmatrix}.$$

a) Show that *another* placement of dominoes also yields the same matrix of numbers.
b) What domino placement yields the array
$$\begin{pmatrix} 3&3&6&5&1&5&1&5\\ 6&5&6&1&2&3&2&4\\ 2&4&3&3&3&6&2&0\\ 4&1&6&1&4&4&6&0\\ 3&0&3&0&1&1&4&4\\ 2&6&2&5&0&5&0&0\\ 2&5&0&5&4&2&1&6 \end{pmatrix}?$$

▶ **143.** [*20*] Show that Dominosa reconstruction is a special case of 3D MATCHING.

144. [*M22*] Generate random instances of Dominosa, and estimate the probability of obtaining a 7×8 matrix with a unique solution. Use two models of randomness: (i) Each matrix whose elements are permutations of the multiset $\{8 \times 0, 8 \times 1, \ldots, 8 \times 6\}$ is equally likely; (ii) each matrix obtained from a random shuffle of the dominoes is equally likely.

▶ **158.** [*20*] Algorithm D can be extended in the following curious way: Let p be the primary item that is covered first, and suppose that there are k ways to cover it. Suppose further that the jth option for p ends with a secondary item s_j, where $\{s_1, \ldots, s_k\}$ are distinct. Modify the algorithm so that, whenever a solution contains the jth option for p, it leaves items $\{s_1, \ldots, s_{j-1}\}$ uncovered. (In other words, the modified algorithm will emulate the behavior of the unmodified algorithm on a much larger instance, in which the jth option for p contains all of $s_1, s_2, \ldots, s_j$.)

▶ **160.** [*25*] Number the options of an exact cover problem from 1 to M. A *minimax solution* is one whose maximum option number is as small as possible. Explain how to modify Algorithm C so that it determines all of the minimax solutions (omitting any that are known to be worse than a solution already found).

161. [*22*] Sharpen the algorithm of exercise 160 so that it produces *exactly one* minimax solution — unless, of course, there are no solutions at all.

164. [*20*] A *double word square* is an $n \times n$ array whose rows and columns contain $2n$ different words. Encode this problem as an exact cover problem with color controls. Can you save a factor of 2 by not generating the transpose of previous solutions? Does

commutative
identity element
Dominosa
solitaire
game
Pijanowski solitaire, see Dominosa
dominoes
3D MATCHING
permutations of the multiset
random domino placement
minimax solution
double word square
word square, double

Algorithm C compete with the algorithm of exercise 7.2.2–28 (which was designed explicitly to handle word-square problems)?

OSPD4
word stair
color controls
NP-complete
2D matching

165. [*21*] Instead of finding *all* of the double word squares, we usually are more interested in finding the *best* one, in the sense of using only words that are quite common. For example, it turns out that a double word square can be made from the words of `WORDS`(1720) but not from those of `WORDS`(1719). Show that it's rather easy to find the smallest W such that `WORDS`(W) supports a double word square, via dancing links.

166. [*24*] What are the best double word squares of sizes 2×2, 3×3, …, 7×7, in the sense of exercise 165, with respect to *The Official* SCRABBLE® *Players Dictionary*? [Exercise 7.2.2–32 considered the analogous problem for *symmetric* word squares.]

▸ **168.** [*22*] A *word stair* of period p is a cyclic arrangement of words, offset stepwise, that contains $2p$ distinct words across and down. They exist in two varieties, left and right:

```
                . . .     . . .
            S T A I R     S T A I R
          S H A R P         S L O O P
        S T E M S             S T O O D
      S C R A P                 S T E E R
    S T A I R                     S T A I R
  S H A R P                         S L O O P
S T E M S             p = 4           S T O O D
S C R A P                               S T E E R
S T A I R                                 S T A I R
. . .                                         . . .
```

What are the best five-letter word stairs, in the sense of exercise 165, for $1 \le p \le 10$? *Hint:* You can save a factor of $2p$ by assuming that the first word is the most common.

169. [*40*] For given W, find the largest p such that `WORDS`(W) supports a word stair of period p. (There are two questions for each W, examining stairs to the {left, right}.)

170. [*24*] Some p-word cycles define *two-way* word stairs that have $3p$ distinct words:

```
                . . .     . . .
            R A P I D     R A P I D
          R A T E D         R A T E D
        L A C E S             L A C E S
      R O B E S                 R O B E S
    R A P I D                     R A P I D
  R A T E D                         R A T E D
L A C E S             p = 4           L A C E S
R O B E S                               R O B E S
R A P I D                                 R A P I D
. . .                                         . . .
```

What are the best five-letter examples of this variety, for $1 \le p \le 10$?

171. [*22*] Another periodic arrangement of $3p$ words, perhaps even nicer than that of exercise 170 and illustrated here for $p = 3$, lets us read them *diagonally* up or down, as well as across. What are the best five-letter examples of *this* variety, for $1 \le p \le 10$? (Notice that there is $2p$-way symmetry.)

```
   . . .
S L A N T  (F L I N T)
F L U N K  (B L A N K)
B L I N K  (S L U N K)
S L A N T  (F L I N T)
F L U N K
B L I N K  (S L I N K)
S L A N T  (F L A N K)
F L U N K  (B L U N T)
   . . .   (S L I N K)
```

175. [*25*] Prove that the exact cover problem with color controls is NP-complete, even if every option consists of only two items.

180. [*22*] Using the "word search puzzle" conventions of Figs. 71 and 72, show that the words `ONE`, `TWO`, `THREE`, `FOUR`, `FIVE`, `SIX`, `SEVEN`, `EIGHT`, `NINE`, `TEN`, `ELEVEN`, and `TWELVE` can all be packed into a 6×6 square, leaving one cell untouched.

181. [*22*] Also pack *two* copies of `ONE`, `TWO`, `THREE`, `FOUR`, `FIVE` into a 5×5 square.

▸ **182.** [*32*] The first 44 presidents of the U.S.A. had 38 distinct surnames: `ADAMS`, `ARTHUR`, `BUCHANAN`, `BUSH`, `CARTER`, `CLEVELAND`, `CLINTON`, `COOLIDGE`, `EISENHOWER`, `FILLMORE`, `FORD`, `GARFIELD`, `GRANT`, `HARDING`, `HARRISON`, `HAYES`, `HOOVER`, `JACKSON`, `JEFFERSON`, `JOHNSON`, `KENNEDY`, `LINCOLN`, `MADISON`, `MCKINLEY`, `MONROE`, `NIXON`, `OBAMA`, `PIERCE`, `POLK`, `REAGAN`, `ROOSEVELT`, `TAFT`, `TAYLOR`, `TRUMAN`, `TYLER`, `VANBUREN`, `WASHINGTON`, `WILSON`.

a) What's the smallest square into which all of these names can be packed, using word search conventions, and requiring all words to be *connected* via overlaps?

b) What's the smallest *rectangle*, under the same conditions?

▸ **183.** [*25*] Pack as many of the following words as possible into a 9×9 array, simultaneously satisfying the rules of *both* word search *and* sudoku:

```
ACRE   COMPARE    CORPORATE   MACRO   MOTET       ROAM
ART    COMPUTER   CROP        META    PARAMETER   TAME
```

▸ **185.** [*28*] A "wordcross puzzle" is the challenge of packing a given set of words into a rectangle under the following conditions: (i) All words must read either across or down, as in a crossword puzzle. (ii) No letters are adjacent unless they belong to one of the given words. (iii) The words are rookwise connected. For example, the eleven words `ZERO`, `ONE`, ..., `TEN` can be placed into an 8×8 square under constraints (i) and (ii) as shown; but (iii) is violated, because there are three different components.

```
T H R E E   F
W         S I X
O N E       V
      S E V E N
  Z           I
  E I G H T   N
  R       E   E
F O U R   N
```

Explain how to encode a wordcross puzzle as an exact cover problem with color controls. Use your encoding to find a correct solution to the problem above. Do those eleven words fit into a *smaller* rectangle, under conditions (i), (ii), and (iii)?

186. [*30*] What's the smallest wordcross square that contains the surnames of the first 44 U.S. presidents? (Use the names in exercise 182, but change `VANBUREN` to `VAN BUREN`.)

187. [*21*] Find all 8×8 crossword puzzle diagrams that contain exactly (a) 12 3-letter words, 12 4-letter words, and 4 5-letter words; (b) 12 5-letter words, 8 2-letter words, and 4 8-letter words. They should have no words of other lengths.

190. [*M25*] Let α be a permutation of the cells of a 9×9 array that takes any sudoku solution into another sudoku solution. We say that α is an *automorphism* of the sudoku solution $S = (s_{ij})$ if there's a permutation π of $\{1, 2, \ldots, 9\}$ such that $s_{(ij)\alpha} = s_{ij}\pi$ for $0 \le i, j < 9$. For example, the permutation that takes ij into $(ij)\alpha = ji$, commonly called transposition, is an automorphism of (28b), with respect to the permutation $\pi = (24)(37)(68)$; but it is *not* an automorphism of (28a) or (28c).

Show that Algorithm C can be used to find all sudoku solutions that have a given automorphism α, by defining an appropriate exact cover problem with color constraints.

How many sudoku solutions have transposition as an automorphism?

191. [*M25*] Continuing exercise 190, how many *hypersudoku* solutions have automorphisms of the following types? (a) transposition; (b) the transformation of exercise 78(d); (c) 90° rotation; (d) both (b) and (c).

word search puzzle
presidents
I'm not sure
how many of
these names
should go in
the index
connected
word search
sudoku
wordcross
crisscross puzzle, composing, see wordcross
rookwise connected
components
crossword puzzle diagrams
5-letter words
permutation
sudoku solution
automorphism
transposition
automorphism
hypersudoku

200. [*M25*] Consider a weighted exact cover problem in which we must choose 2 of 4 options to cover item 1, and 5 of 7 options to cover item 2; the options don't interact.

a) What's the size of the search tree if we branch first on item 1, then on item 2? Would it better to branch first on item 2, then on item 1?

b) Generalize part (a) to the case when item 1 needs p of $p+d$ options, while item 2 needs q of $q+d$ options, where $q > p$ and $d > 0$.

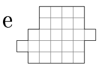
weighted exact cover problem

EXERCISES — Second Set

Hundreds of fascinating recreational problems have been based on polyominoes and their polyform cousins (the polycubes, polyiamonds, polyhexes, polysticks, ...). The following exercises explore "the cream of the crop" of such classic puzzles, as well as a few gems that were not discovered until recently.

In most cases the point of the exercise is to find a good way to discover all solutions, usually by setting up an appropriate exact cover problem that can be solved without taking an enormous amount of time.

▶ **240.** [*25*] Sketch the design of a utility program that will create sets of options by which an exact cover solver will fill a given shape with a given set of polyominoes.

248. [*18*] Using Conway's piece names, pack five pentominoes into the shape  so that they spell a common English word when read from left to right.

▶ **250.** [*21*] There are 1010 ways to pack the twelve pentominoes into a 5×12 box, not counting reflections. What's a good way to find them all, using Algorithm D?

251. [*21*] How many of those 1010 packings decompose into $5 \times k$ and $5 \times (12-k)$?

252. [*21*] In how many ways can the eleven nonstraight pentominoes be packed into a 5×11 box, not counting reflections? (Reduce symmetry cleverly.)

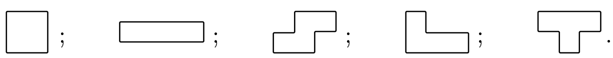
254. [*20*] There are 2339 ways to pack the twelve pentominoes into a 6×10 box, not counting reflections. What's a good way to find them all, using Algorithm D?

255. [*23*] Continuing exercise 254, explain how to find special kinds of packings:
a) Those that decompose into $6 \times k$ and $6 \times (10-k)$.
b) Those that have all twelve pentominoes touching the outer boundary.
c) Those with all pentominoes touching that boundary *except* for V, which doesn't.
d) Same as (c), with each of the other eleven pentominoes in place of V.
e) Those with the *minimum* number of pentominoes touching the outer boundary.
f) Those that are characterized by Arthur C. Clarke's description, as quoted below. (That is, the X pentomino should touch only the F, N, U, and V — no others.)

Very gently, he replaced the titanite cross
in its setting between the F, N, U, and V pentominoes.
— ARTHUR C. CLARKE, *Imperial Earth* (1976)

256. [*25*] All twelve pentominoes fit into a 3×20 box only in two ways, shown in (36).
a) How many ways are there to fit *eleven* of them into that box?
b) In how many solutions to (a) are the five holes *nonadjacent*, kingwise?
c) In how many ways can eleven pentominoes be packed into a 3×19 box?

257. [*21*] There are five different *tetrominoes*, namely



square; straight; skew; ell; tee.

In how many essentially different ways can each of them be packed into an 8×8 square together with the twelve pentominoes?

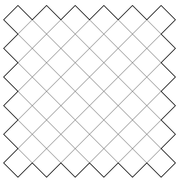
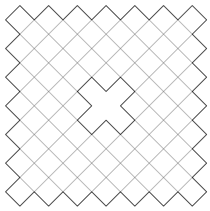
258. [*21*] If an 8×8 checkerboard is cut up into thirteen pieces, representing the twelve pentominoes together with one of the tetrominoes, some of the pentominoes will have more black cells than white. Is it possible to do this in such a way that U, V, W, X, Y, Z have a black majority while the others do not?

259. [*18*] Design a nice, simple tiling pattern that's based on the five tetrominoes.

polyform
Conway
five-letter words
pentominoes
nonstraight
symmetry
pentominoes
pentominoes, names of
CLARKE
tetrominoes
tetrominoes, names of
tetrominoes

260. [*25*] How many of the 6×10 pentomino packings are *strongly three-colorable*, in the sense that each individual piece could be colored red, white, or blue in such a way that no pentominoes of the same color touch each other — not even at corner points?

▸ **262.** [*20*] The black cells of a square $n \times n$ checkerboard form an interesting graph called the *Aztec diamond* of order $n/2$. For example, the cases $n = 11$ and 13 are illustrated by the outer boundaries of

(i)  and (ii)  ,

where (ii) has a "hole" showing the case $n = 3$. Thus (i) has 61 cells, and (ii) has 80.

a) Find all ways to pack (i) with the twelve pentominoes and one monomino.

b) Find all ways to pack (ii) with the $12 + 5$ pentominoes and tetrominoes.

Speed up the process by not producing solutions that are symmetric to each other.

▸ **263.** [*M26*] Arrange the twelve pentominoes into a Möbius strip of width 4. The pattern should be "faultfree": Every straight line must intersect some piece.

264. [*40*] (H. D. Benjamin, 1948.) Show that the twelve pentominoes can be wrapped around a cube of size $\sqrt{10} \times \sqrt{10} \times \sqrt{10}$. For example, here are front and back views of such a cube, made from twelve colorful fabrics by the author's wife in 1993:

(Photos by Hector Garcia)

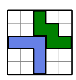
What is the best way to do this, minimizing undesirable distortions at the corners?

▸ **265.** [*22*] (Craig S. Kaplan.) A polyomino can sometimes be surrounded by non-overlapping copies of itself that form a *fence*: Every cell that touches the polyomino — even at a corner — is part of the fence; conversely, every piece of the fence touches the inner polyomino. Furthermore, the pieces must not enclose any unoccupied "holes."

Find the (a) smallest and (b) largest fences for each of the twelve pentominoes. (Some of these patterns are unique, and quite pretty.)

266. [*22*] Solve exercise 265 for fences that satisfy the *tatami* condition of exercise 7.1.4–215: No four edges of the tiles should come together at any "crossroads."

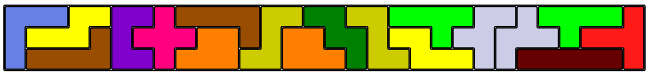
▸ **267.** [*27*] Solomon Golomb discovered in 1965 that there's only one placement of two pentominoes in a 5×5 square that blocks the placement of all the others.



Place (a) $\{I, P, U, V\}$ and (b) $\{F, P, T, U\}$ into a 7×7 square in such a way that none of the other eight will fit in the remaining spaces.

three-colorable
graph coloring
checkerboard
Aztec diamond
symmetric
Möbius strip
faultfree
Benjamin
cube, wrapped
Knuth, Jill
Garcia, Hector
Kaplan
fence
holes
tatami
crossroads
Golomb

268. [*21*] (T. H. O'Beirne, 1961.) The *one-sided pentominoes* are the eighteen distinct 5-cell pieces that can arise if we aren't allowed to flip pieces over:



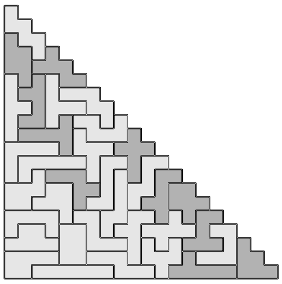
Notice that there now are two versions of F, L, P, N, Y, and Z.

In how many ways can all eighteen of them be packed into rectangles?

269. [*21*] Suppose you want to pack the twelve pentominoes into a 6×10 box, *without* turning any pieces over. Then 2^6 different problems arise, depending on which sides of the one-sided pieces are present. Which of those 64 problems has (a) the fewest (b) the most solutions?

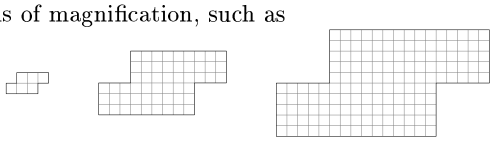
270. [*21*] When tetrominoes are both checkered and one-sided (see exercises 258 and 268), ten possible pieces arise. In how many ways can all ten of them fill a rectangle?

275. [*20*] There are 35 *hexominoes*, first enumerated in 1934 by the master puzzlist H. D. Benjamin. At Christmastime that year, he offered ten shillings to the first person who could pack them into a 14×15 rectangle — although he wasn't sure whether or not it could be done. The prize was won by F. Kadner, who proved that the hexominoes actually *can't* be packed into *any* rectangle. Nevertheless, Benjamin continued to play with them, eventually discovering that they fit nicely into the triangle shown here.



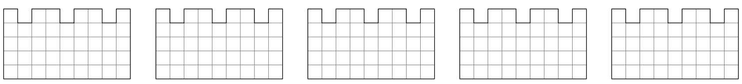
Prove Kadner's theorem. *Hint:* See exercise 258.

276. [*24*] (Frans Hansson, 1947.) The fact that $35 = 1^2 + 3^2 + 5^2$ suggests that we might be able to pack the hexominoes into three boxes that represent a *single* hexomino shape at three levels of magnification, such as

 .

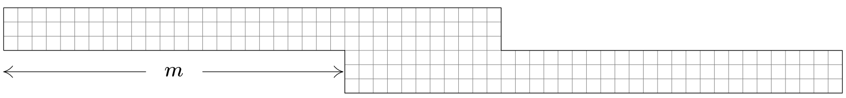
For which hexominoes can this be done?

▸ **277.** [*30*] Show that the 35 hexominoes can be packed into five "castles":

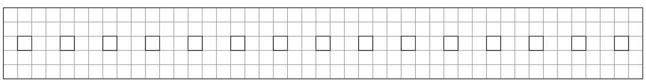
 .

In how many ways can this be done?

278. [*41*] For which values of m can the hexominoes be packed into a box like this?



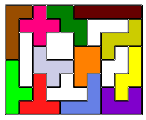
279. [*41*] Perhaps the nicest hexomino packing uses a 5×45 rectangle with 15 holes

 ,

proposed by W. Stead in 1954. In how many ways can the 35 hexominoes fill it?

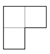
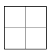
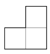
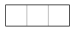
O'Beirne
one-sided pentominoes
flip pieces over
tetrominoes
checkered
one-sided
checkerboard dissections
hexominoes
Benjamin
Kadner
Hansson
magnification
triplication
castles
Stead

▸ **281.** [*22*] In how many ways can the twelve pentominoes be placed into an 8×10 rectangle, leaving holes in the shapes of the five tetrominoes? (The holes should not touch the boundary, nor should they touch each other, even at corners; one example is shown at the right.) Explain how to encode this puzzle as an exact cover problem with color controls.



282. [*46*] If possible, solve the analog of exercise 281 for the case of 35 *hexominoes* in a 5×54 rectangle, leaving holes in the shapes of the twelve *pentominoes*.

▸ **298.** [*HM35*] A *parallelogram polyomino*, or "parallomino" for short, is a polyomino whose boundary consists of two paths that each travel only north and/or east. (Equivalently, it is a "skew Young tableau" or a "skew Ferrers board," the difference between the diagrams of two tableaux or partitions; see Sections 5.1.4 and 7.2.1.4.) For example, there are five parallominoes whose boundary paths have length 4:

NNNE ENNN  ; NNEE ENEN  ; NNEE EENN  ; NENE EENN  ; NEEE EEEN  .

a) Find a one-to-one correspondence between the set of ordered trees with m leaves and n nodes and the set of parallominoes with width m and height $n-m$. The area of each parallomino should be the path length of its corresponding tree.

b) Study the generating function $G(w,x,y) = \sum_{\text{parallominoes}} w^{\text{area}} x^{\text{width}} y^{\text{height}}$.

c) Prove that the parallominoes whose width-plus-height is n have total area 4^{n-2}.

d) Part (c) suggests that we might be able to pack all of those parallominoes into a $2^{n-2} \times 2^{n-2}$ square, *without* rotating them or flipping them over. Such a packing is clearly impossible when $n = 3$ or $n = 4$; but is it possible when $n = 5$ or $n = 6$?

300. [*20*] Extend exercise 240 to three dimensions. How many base placements do each of the seven Soma pieces have?

▸ **302.** [*22*] The *Somap* is the graph whose vertices are the 240 distinct solutions to the Soma cube problem, with $u — v$ if and only if u can be obtained from v by changing the positions of at most three pieces. (Using the terminology of answer 31(e), adjacent vertices correspond to solutions of *semidistance* ≤ 3.) The *strong* Somap is similar, but it has $u — v$ only when a change of just *two* pieces gets from one to the other.

a) What are the degree sequences of these graphs?

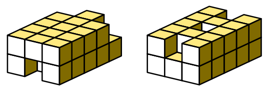
b) How many connected components do they have? How many bicomponents?

303. [*M21*] If a $(3m+1) \times (3n+2)$ box is packed with $3mn+2m+n$ straight trominoes and one domino, where must the domino be placed?

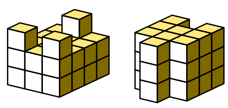
▸ **304.** [*M25*] Use factorization to prove that Fig. 80's W-wall cannot be built.

305. [*24*] Figure 80(a) shows some of the many "low-rise" (2-level) shapes that can be built from the seven Soma pieces. Which of them is hardest (has the fewest solutions)? Which is easiest? Answer these questions also for the 3-level prism shapes in Fig. 80(b).

▸ **306.** [*M23*] Generalizing the first four examples of Fig. 80, study the set of *all* shapes obtainable by deleting three cubies from a $3 \times 5 \times 2$ box. (Two examples are shown here.) How many essentially different shapes are possible? Which shape is easiest? Which shape is hardest?



307. [*22*] Similarly, consider (a) all shapes that consist of a $3 \times 4 \times 3$ box with just three cubies in the top level; (b) all 3-level prisms that fit into a $3 \times 4 \times 3$ box.



pentominoes
tetrominoes
color controls
hexominoes
parallelogram polyomino
parallomino
skew Young tableau
Young tableaux
skew Ferrers board
Ferrers diagrams
tableaux
partitions
trees
path length
generating function
base placements
Somap
Soma cube
semidistance
degree sequences
connected components
bicomponents
straight trominoes
domino
factorization
W-wall
Soma pieces

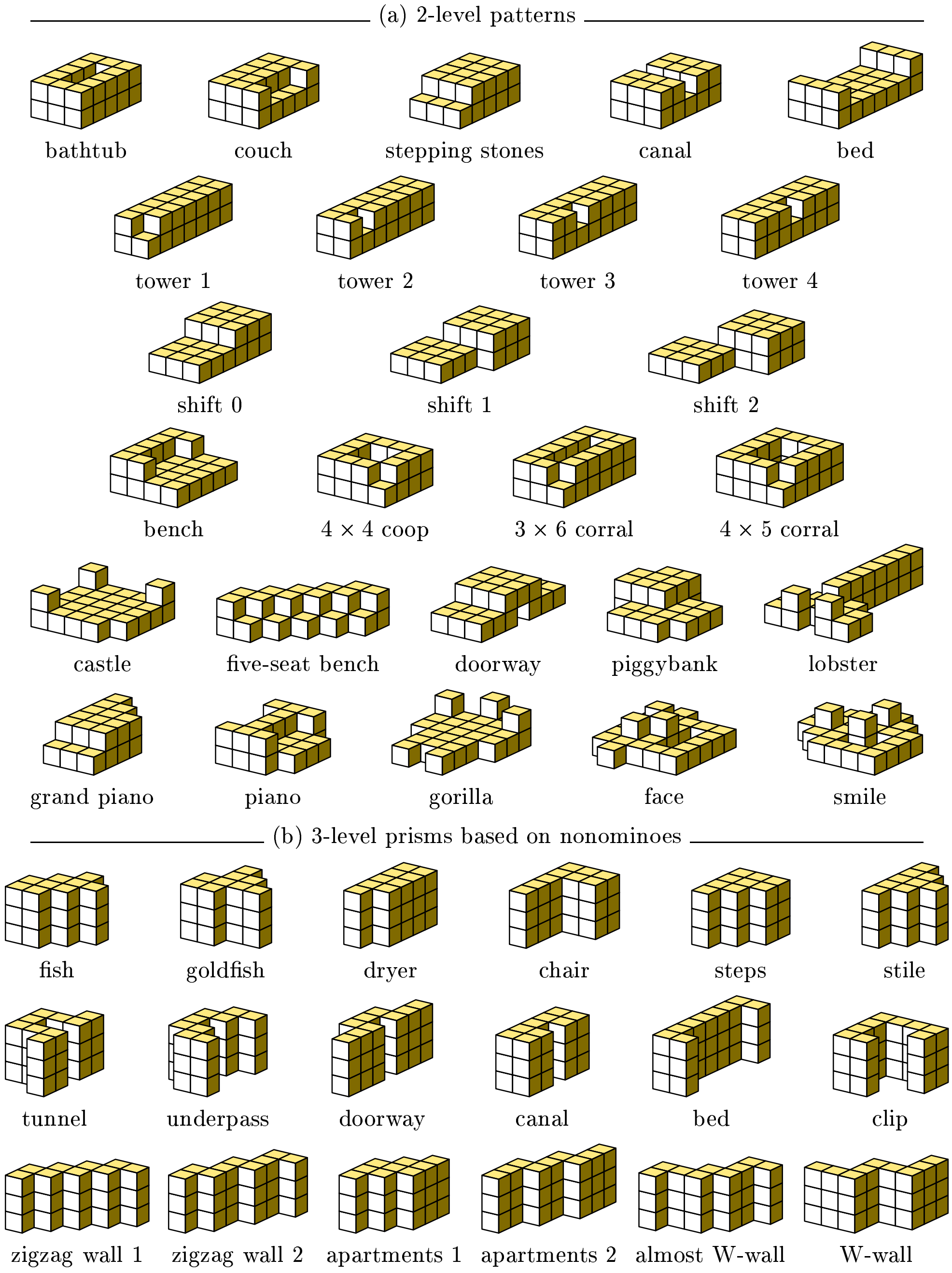
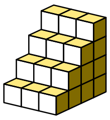


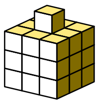
Fig. 80. Gallery of noteworthy polycubes that contain 27 cubies. All of them can be built from the seven Soma pieces, except for the W-wall. Many constructions are also stable when tipped on edge and/or when turned upside down. (See exercises 304–314.)

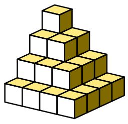
308. [*25*] How many of the 1285 *nonominoes* define a prism that can be realized by the Soma pieces? Do any of those packing problems have a unique solution?

310. [*M40*] Make empirical tests of Piet Hein's belief that the number of shapes achievable with seven Soma pieces is approximately the number of 27-cubie polycubes.

312. [*20*] (B. L. Schwartz, 1969.) Show that the Soma pieces can make shapes that appear to have more than 27 cubies, because of holes hidden inside or at the bottom:

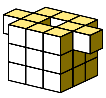

staircase

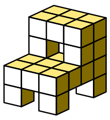

penthouse

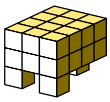

pyramid

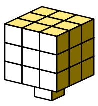
In how many ways can these three shapes be constructed?

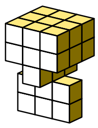
313. [*22*] Show that the seven Soma pieces can also make structures such as


casserole


cot

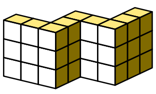

vulture

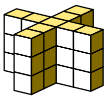

mushroom

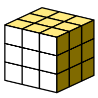

cantilever ,

which are "self-supporting" via gravity. (You may need to place a small book on top.)

▶ **314.** [*M32*] Impossible structures *can* be built, if we insist only that they look genuine when viewed from the front (like façades in Hollywood movies)! Find all solutions to


W-wall


X-wall

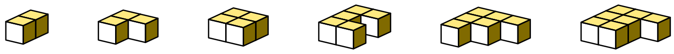

cube

that are visually correct. (To solve this exercise, you need to know that the illustrations here use the non-isometric projection $(x, y, z) \mapsto (30x - 42y, 14x + 10y + 45z)u$ from three dimensions to two, where u is a scale factor.) All seven Soma pieces must be used.

nonominoes
Hein
Schwartz
self-supporting
gravity
façades
movies
isometric
projection
three dimensions

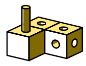
Cube Diabolique
Diabolical Cube
Watilliaux
L-bert Hall
holes
dowel
tetracubes
gravity
octominoes
pentacubes
solid pentominoes
flat pentacubes
mirror images
pentominoes
$5 \times 5 \times 5$ cube
Dowler's Box

315. [*30*] The earliest known example of a polycube puzzle is the "Cube Diabolique," manufactured in late nineteenth century France by Charles Watilliaux; it contains six flat pieces of sizes 2, 3, . . . , 7:

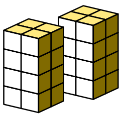


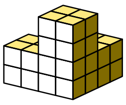
a) In how many ways do these pieces make a $3 \times 3 \times 3$ cube?
b) Are there six polycubes, of sizes 2, 3, . . . , 7, that make a cube in just *one* way?

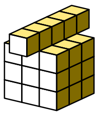
316. [*21*] (*The L-bert Hall.*) Take two cubies and drill three holes through each of them; then glue them together and attach a solid cubie and dowel, as shown. Prove that there's only one way to pack nine such pieces into a $3 \times 3 \times 3$ box.

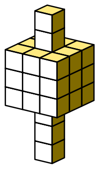


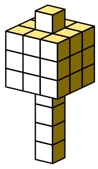
317. [*22*] Show that there are exactly eight different *tetracubes* — polycubes of size 4. Which of the following shapes can they make, respecting gravity? How many solutions are possible?

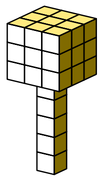

twin towers


double claw


cannon

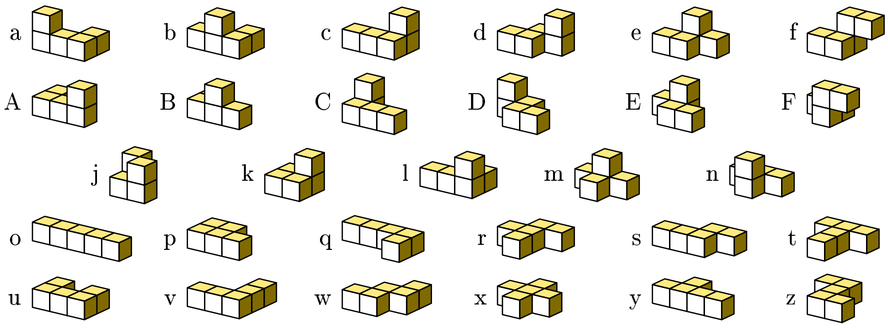

up 3


up 4


up 5

318. [*25*] How many of the 369 *octominoes* define a 4-level prism that can be realized by the tetracubes? Do any of those packing problems have a unique solution?

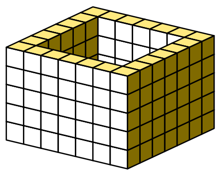
320. [*30*] There are 29 *pentacubes*, conveniently identified with one-letter codes:



Pieces o through z are called, not surprisingly, the *solid pentominoes* or *flat pentacubes*.

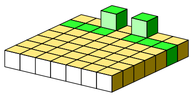
a) What are the mirror images of a, b, c, d, e, f, A, B, C, D, E, F, j, k, l, . . . , z?
b) In how many ways can the solid pentominoes be packed into an $a \times b \times c$ cuboid?
c) What "natural" set of 25 pentacubes is able to fill the $5 \times 5 \times 5$ cube?

▸ **321.** [*25*] The full set of 29 pentacubes can build an enormous variety of elegant structures, including a particularly stunning example called "Dowler's Box." This $7 \times 7 \times 5$ container, first considered by R. W. M. Dowler in 1979, is constructed from five flat slabs. Yet only 12 of the pentacubes lie flat; the other 17 must somehow be worked into the edges and corners.

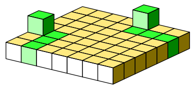


Despite these difficulties, Dowler's Box has so many solutions that we can actually impose many further conditions on its construction:

a) Build Dowler's Box in such a way that the chiral pieces a, b, c, d, e, f and their images A, B, C, D, E, F all appear in horizontally mirror-symmetric positions. chiral mirror



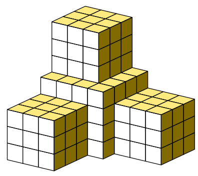
horizontally symmetric c and C



diagonally symmetric c and C

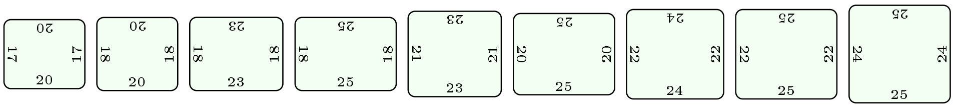
b) Alternatively, build it so that those pairs are *diagonally* mirror-symmetric.

c) Alternatively, place piece x in the center, and build the remaining structure from four congruent pieces that have seven pentacubes each.

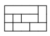


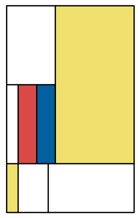
322. [*25*] The 29 pentacubes can also be used to make the shape shown here, exploiting the curious fact that $3^4 + 4^3 = 29 \cdot 5$. But Algorithm D will take a long, long time before telling us how to construct it, unless we're lucky, because the space of possibilities is huge. How can we find a solution quickly?

339. [*29*] Nick Baxter devised an innocuous-looking but maddeningly difficult "Square Dissection" puzzle for the International Puzzle Party in 2014, asking that the nine pieces



be placed flat into a 65×65 square. One quickly checks that $17\times 20+18\times 20+\cdots+24\times 25 = 65^2$; yet nothing seems to work! Solve his puzzle with the help of Algorithm D.

▶ **340.** [*20*] The next group of exercises is devoted to the decomposition of rectangles into rectangles, as in the Mondrianesque pattern shown here. The *reduction* of such a pattern is obtained by distorting it, if necessary, so that it fits into an $m \times n$ grid, with each of the vertical coordinates $\{0, 1, \ldots, m\}$ used in at least one horizontal boundary and each of the horizontal coordinates $\{0, 1, \ldots, n\}$ used in at least one vertical boundary. For example, the illustrated pattern reduces to , where $m = 3$ and $n = 5$. (Notice that the original rectangles needn't have rational width or height.)



A pattern is called *reduced* if it is equal to its own reduction. Design an exact cover problem by which Algorithm M will discover all of the reduced decompositions of an $m\times n$ rectangle, given m and n. How many of them are possible when $(m, n) = (3, 5)$?

341. [*M25*] The maximum number of subrectangles in a reduced $m \times n$ pattern is obviously mn. What is the *minimum* number?

342. [*10*] A reduced pattern is called *strictly reduced* if each of its subrectangles $[a\,.\,.\,b)\times[c\,.\,.\,d)$ has $(a, b) \neq (0, m)$ and $(c, d) \neq (0, n)$ — in other words, if no subrectangle "cuts all the way across." Modify the construction of exercise 340 so that it produces only strictly reduced solutions. How many 3×5 patterns are strictly reduced?

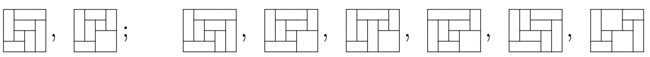
343. [*20*] A rectangle decomposition is called *faultfree* if it cannot be split into two or more rectangles. For example,  is *not* faultfree, because it has a fault line between rows 2 and 3. (It's easy to see that every reduced faultfree pattern is *strictly* reduced, unless $m = n = 1$.) Modify the construction of exercise 340 so that it produces only faultfree solutions. How many reduced 3×5 patterns are faultfree?

344. [*23*] True or false: Every faultfree packing of an $m\times n$ rectangle by 1×3 trominoes is reduced, except in the trivial cases $(m, n) = (1, 3)$ or $(3, 1)$.

347. [*22*] (*Motley dissections.*) Many of the most interesting decompositions of an $m\times n$ rectangle involve strictly reduced patterns whose subrectangles $[a_i\,.\,.\,b_i)\times[c_i\,.\,.\,d_i)$ satisfy the extra condition

$$(a_i, b_i) \neq (a_j, b_j) \quad\text{and}\quad (c_i, d_i) \neq (c_j, d_j) \quad\text{when } i < j.$$

Thus no two subrectangles are cut off by the same pair of horizontal or vertical lines. The smallest such "motley dissections" are the 3×3 pinwheels,  and , which are considered to be essentially the same because they are mirror images of each other. There are eight essentially distinct motley rectangles of size $4 \times n$, namely



The two 4×4s can each be drawn in 8 different ways, under rotations and reflections. Similarly, most of the 4×5s can be drawn in 4 different ways. But the last two have only two forms, because they're symmetric under 180° rotation.

Baxter
Square Dissection
rectangles into rectangles
Mondrian
reduction
strictly reduced
faultfree
trominoes
straight trominoes: 1×3
Motley dissections
pinwheels
rotations and reflections
symmetric under 180° rotation

Design an exact cover problem by which Algorithm M will discover all of the motley dissections of an $m \times n$ rectangle, given m and n. (When $m = n = 4$ the algorithm should find $8 + 8$ solutions; when $m = 4$ and $n = 5$ it should find $4 + 4 + 4 + 4 + 2 + 2$.)

▸ **348.** [*25*] Improve the construction of the previous exercise by taking advantage of symmetry to cut the number of solutions in half. (When $m = 4$ there will now be $4 + 4$ solutions when $n = 4$, and $2 + 2 + 2 + 2 + 1 + 1$ when $n = 5$.) *Hint:* A motley dissection is never identical to its left-right reflection, so we needn't visit both.

349. [*20*] The *order* of a motley dissection is the number of subrectangles it has. There are no motley dissections of order six. Show, however, that there are $m \times m$ motley dissections of order $2m - 1$ and $m \times (m+1)$ motley dissections of order $2m$, for all $m > 3$.

350. [*21*] An $m \times n$ motley dissection must have order less than $\binom{m+1}{2}$, because only $\binom{m+1}{2} - 1$ intervals $[a_i \,..\, b_i)$ are permitted. What is the maximum order that's actually achievable by an $m \times n$ motley dissection, for $m = 5$, 6, and 7?

▸ **352.** [*23*] Explain how to generate all of the $m \times n$ motley dissections that have 180°-rotational symmetry, as in the last two examples of exercise 347, by modifying the construction of exercise 348. (In other words, if $[a \,..\, b) \times [c \,..\, d)$ is a subrectangle of the dissection, its complement $[m - b \,..\, m - a) \times [n - d \,..\, n - c)$ must also be one of the subrectangles, possibly the same one.) How many such dissections have size 8×16?

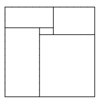
353. [*24*] Further symmetry is possible when $m = n$ (as in exercise 347's pinwheel).

a) Explain how to generate all of the $n \times n$ motley dissections that have 90°-rotational symmetry. This means that $[a \,..\, b) \times [c \,..\, d)$ implies $[c \,..\, d) \times [n{-}b \,..\, n{-}a)$.

b) Explain how to generate all of the $n \times n$ motley dissections that are symmetric under reflection about both diagonals. This means that $[a \,..\, b) \times [c \,..\, d)$ implies $[c \,..\, d) \times [a \,..\, b)$ and $[n{-}d \,..\, n{-}c) \times [n{-}b \,..\, n{-}a)$, hence $[n{-}b \,..\, n{-}a) \times [n{-}d \,..\, n{-}c)$.

c) What's the smallest n for which symmetric solutions of type (b) exist?

355. [*26*] A "perfectly decomposed rectangle" of order t is a dissection of a rectangle into t subrectangles $[a_i \,..\, b_i) \times [c_i \,..\, d_i)$ such that the $2t$ dimensions $b_1 - a_1$, $d_1 - c_1$, …, $b_t - a_t$, $d_t - c_t$ are all distinct. For example, five rectangles of sizes 1×2, 3×7, 4×6, 5×10, and 8×9 can be assembled to make the perfectly decomposed 13×13 square shown here. What are the *smallest possible* perfectly decomposed squares of orders 5, 6, 7, 8, 9, and 10, having integer dimensions?



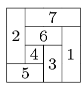
356. [*M28*] An "incomparable dissection" of order t is a decomposition of a rectangle into t subrectangles, none of which will fit inside another. In other words, if the widths and heights of the subrectangles are respectively $w_1 \times h_1$, …, $w_t \times h_t$, we have neither ($w_i \le w_j$ and $h_i \le h_j$) nor ($w_i \le h_j$ and $h_i \le w_j$) when $i \ne j$.

a) True or false: An incomparable dissection is perfectly decomposed.

b) True or false: The reduction of an incomparable dissection is motley.

c) True or false: The reduction of an incomparable dissection can't be a pinwheel.

d) Prove that every incomparable dissection of order ≤ 7 reduces to the first 4×4 motley dissection in exercise 347. Furthermore its seven regions can be labeled as shown, with $w_1 < w_2 < \cdots < w_6 < w_7$ and $h_7 < h_6 < \cdots < h_2 < h_1$.



e) Suppose the reduction of an incomparable dissection is $m \times n$, and suppose its regions have been labeled $\{1, \ldots, t\}$. Then there are numbers $x_1, \ldots, x_n, y_1, \ldots, y_m$ such that the widths are sums of the x's and the heights are sums of the y's. (For example, in (d) we have $w_2 = x_1$, $h_2 = y_1 + y_2 + y_3$, $w_7 = x_2 + x_3 + x_4$, $h_7 = y_1$, etc.) Prove that such a dissection exists with $w_1 < w_2 < \cdots < w_t$ if and only if the

symmetry
order
180°-rotational symmetry
complement
pinwheel
90°-rotational symmetry
reflection about both diagonals
bidiagonal symmetry
perfectly decomposed rectangle
incomparable dissection
motley
reduction

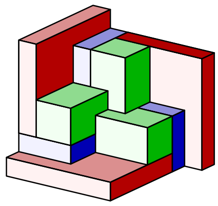
linear inequalities $w_1 < w_2 < \cdots < w_t$ have a positive solution $(x_1, \ldots, x_n)$ and the linear inequalities $h_1 > h_2 > \cdots > h_t$ have a positive solution $(y_1, \ldots, y_m)$.

linear inequalities
subcuboids
Kim

357. [*M29*] Among all the incomparable dissections of order (a) seven and (b) eight, restricted to integer sizes, find the rectangles with smallest possible perimeter. Also find the smallest possible *squares* that have incomparable dissections in integers. *Hint:* Show that there are 2^t potential ways to mix the w's with the h's, preserving their order; and find the smallest perimeter for each of those cases.

▸ **358.** [*M25*] Find seven *different* rectangles of area 1/7 that can be assembled into a square of area 1, and prove that the answer is unique.

▸ **360.** [*18*] There's a natural way to extend the idea of motley dissection to three dimensions, by subdividing an $l \times m \times n$ cuboid into subcuboids $[a_i \,..\, b_i) \times [c_i \,..\, d_i) \times [e_i \,..\, f_i)$ that have no repeated intervals $[a_i \,..\, b_i)$ or $[c_i \,..\, d_i)$ or $[e_i \,..\, f_i)$.



For example, Scott Kim has discovered a remarkable motley $7 \times 7 \times 7$ cube consisting of 23 individual blocks, 11 of which are illustrated here. (Two of them are hidden behind the others.) The full cube is obtained by suitably placing a mirror image of these pieces in front, together with a $1 \times 1 \times 1$ cubie in the center.

By studying this picture, show that Kim's construction can be defined by coordinate intervals $[a_i \,..\, b_i) \times [c_i \,..\, d_i) \times [e_i \,..\, f_i)$, with $0 \le a_i, b_i, c_i, d_i, e_i, f_i \le 7$ for $1 \le i \le 23$, in such a way that the pattern is symmetrical under the transformation $xyz \mapsto \bar{y}\bar{z}\bar{x}$. In other words, if $[a \,..\, b) \times [c \,..\, d) \times [e \,..\, f)$ is one of the subcuboids, so is $[7-d \,..\, 7-c) \times [7-f \,..\, 7-e) \times [7-b \,..\, 7-a)$.

361. [*29*] Use exercise 360 to construct a perfectly decomposed $108 \times 108 \times 108$ cube, consisting of 23 subcuboids that have 69 distinct integer dimensions. (See exercise 356.)

362. [*24*] By generalizing exercises 347 and 348, explain how to find *every* dissection of an $l \times m \times n$ cuboid, using Algorithm M. *Note:* In three dimensions, the strictness condition '$(a_i, b_i) \ne (0, m)$ and $(c_i, d_i) \ne (0, n)$' of exercise 342 should become

$$\bigl[(a_i, b_i) = (0, l)\bigr] + \bigl[(c_i, d_i) = (0, m)\bigr] + \bigl[(e_i, f_i) = (0, n)\bigr] \;\le\; 1.$$

What are the results when $l = m = n = 7$?

363. [*M46*] Do motley cuboids of size $l \times m \times n$ exist only when $l = m = n = 7$?

999. [*M00*] this is another temporary exercise (for dummies)

Pell
AUBREY
Catalan number
topological sortings
partial order
degenerate
height
bipartite
hitting set problem
vertex cover problem
Secondary

Dr Pell was wont to say, that in the Resolution of Questiones,
the main matter is the well stating them:
which requires a good mother-witt & Logick: as well as Algebra:
for let the Question be but well-stated, and it will worke of it selfe:
. . . By this way, an man cannot intangle his notions, & make a false Steppe.
— JOHN AUBREY, *An Idea of Education of Young Gentlemen* (c. 1684)

SECTION 7.2.2.1

1. (a) Note first that Algorithm 6.2.2T has its own `LLINK` and `RLINK` fields, for left and right children; they shouldn't be confused with the links of the doubly linked list. After all deletions are done, `LLINK`(k) will be the largest search-tree ancestor of k that's less than k; `RLINK`(k) will be the smallest ancestor of k that's greater than k; but if there's no such ancestor, the link will be 0. (For example, in Fig. 10 of Section 6.2.2, `RLINK(LEO)` would be `PISCES` and `LLINK(AQUARIUS)` would be the list head.)

(b) There are $C_n = \binom{2n}{n}\frac{1}{n+1}$ classes (the Catalan number), one for each binary tree.

(c) The size of each class is the number of topological sortings of the partial order generated by the relations $k \prec$ `LLINK`(k), $k \prec$ `RLINK`(k). And this number equals 1 only in the 2^{n-1} "degenerate" trees of height n (see exercise 6.2.2–5).

2. (a) Let `L` and `R` denote the binary tree links. Operation (2) changes `RLINK`(k) only if we are undeleting j with `LLINK`$(j) = k$. If no such elements exist, we have `R`$(k) = \Lambda$, and `RLINK`(k) was never changed by any deletion. Otherwise those elements are $\{j_1, \ldots, j_t\}$, where `R`$(k) = j_1$, `L`$(j_i) = j_{i+1}$, and `L`$(j_t) = \Lambda$. Hence $j_t = k + 1$. Undeleting j_i will set `LLINK`$(j_{i-1}) \leftarrow j_i$, for $i = t, t-1, \ldots, 2$; this leaves `RLINK`$(k) = k + 1$. A similar argument works for `LLINK`, and for links involving the list head.

(Programmers are advised *not* to use the amazing fact just proved, because the lists are malformed during the process; they're fully reconstructed only at the end.)

(b) No. For example, delete 1, 2, 3; then undelete 1, 3, 2.

3. (a) $(x_1, \ldots, x_6) = (1, 0, 0, 1, 1, 0)$. (In general the solutions to linear equations won't always be 0 or 1. For example, the equations $x_1+x_2 = x_2+x_3 = x_1+x_3 = 1$ imply that $x_1 = x_2 = x_3 = \frac{1}{2}$; hence the corresponding exact cover problem is unsolvable.)

(b) In practice, m is much larger than n. Example (5) is just a "toy problem"! The best we can hope to achieve from n simultaneous equations is to express n of the variables in terms of the other $m - n$; that leaves 2^{m-n} cases to try.

4. If G is bipartite, the exact covers are the ways to choose the vertices of one part. (Hence there are 2^k solutions, if G has k components.) Otherwise there are no solutions. (Algorithm D will discover that fact quickly, although Algorithm 7B is faster.)

5. Given a hypergraph, find a set of vertices that hits each hyperedge exactly once. (In an ordinary graph this is the scenario of exercise 4.)

Similarly, the so-called hitting set problem is dual to the vertex cover problem.

6. The header nodes, numbered 1 through N, are followed by L ordinary nodes and $M+1$ spacers; hence the final node Z is number $L+M+N+1$. (There also are $N+1$ records for the horizontal list of items; those "records" aren't true "nodes.")

7. Node 23 is a spacer; '-4' indicates that it follows the 4th option. (Any nonpositive number would work, but this convention aids debugging.) Option 5 ends at node 25.

8. (*Secondary* items, which are introduced in the text after (24), are also handled by the steps below. Such items should be named after all primary items on the first line, and separated from them by some distinguishing mark.)

I1. [Read the first line.] Set $i \leftarrow N_1 \leftarrow 0$. Then, for each item name α on the first line, set $i \leftarrow i+1$, $\mathtt{NAME}(i) \leftarrow \alpha$, $\mathtt{LLINK}(i) \leftarrow i-1$, $\mathtt{RLINK}(i-1) \leftarrow i$. If α names the first secondary item, also set $N_1 \leftarrow i-1$. (In practice α is limited to at most 8 characters, say. One should report an error if $\alpha = \mathtt{NAME}(j)$ for some $j < i$.) spacer

I2. [Finish the horizontal list.] Set $N \leftarrow i$. If $N_1 = 0$ (there were no secondary items), set $N_1 \leftarrow N$. Then set $\mathtt{LLINK}(N+1) \leftarrow N$, $\mathtt{RLINK}(N) \leftarrow N+1$, $\mathtt{LLINK}(N_1+1) \leftarrow N+1$, $\mathtt{RLINK}(N+1) \leftarrow N_1+1$, $\mathtt{LLINK}(0) \leftarrow N_1$, $\mathtt{RLINK}(N_1) \leftarrow 0$. (The active secondary items, if any, are accessible from record $N+1$.)

I3. [Prepare for options.] Set $\mathtt{LEN}(i) \leftarrow 0$ and $\mathtt{ULINK}(i) \leftarrow \mathtt{DLINK}(i) \leftarrow i$ for $1 \le i \le N$. (These are the header nodes for the N item lists.) Then set $M \leftarrow 0$, $p \leftarrow N+1$, $\mathtt{TOP}(p) \leftarrow 0$. (Node p is the first spacer.)

I4. [Read an option.] Terminate with $Z \leftarrow p$ if no input remains. Otherwise let the next line of input contain the item names $\alpha_1 \;\ldots\; \alpha_k$, and do the following for $1 \le j \le k$: Use an algorithm from Chapter 6 to find the index i_j for which $\mathtt{NAME}(i_j) = \alpha_j$. (Report an error if unsuccessful. Complain also if an item name appears more than once in the same option, because a duplicate might make Algorithm D fail spectacularly.) Set $\mathtt{LEN}(i_j) \leftarrow \mathtt{LEN}(i_j)+1$, $q \leftarrow \mathtt{ULINK}(i_j)$, $\mathtt{ULINK}(p+j) \leftarrow q$, $\mathtt{DLINK}(q) \leftarrow p+j$, $\mathtt{DLINK}(p+j) \leftarrow i_j$, $\mathtt{ULINK}(i_j) \leftarrow p+j$, $\mathtt{TOP}(p+j) \leftarrow i_j$.

I5. [Finish an option.] Set $M \leftarrow M+1$, $\mathtt{DLINK}(p) \leftarrow p+k$, $p \leftarrow p+k+1$, $\mathtt{TOP}(p) \leftarrow -M$, $\mathtt{ULINK}(p) \leftarrow p-k$, and return to step I4. (Node p is the next spacer.) ▌

9. Set $\theta \leftarrow \infty$, $p \leftarrow \mathtt{RLINK}(0)$. While $p \ne 0$, do the following: Set $l \leftarrow \mathtt{LEN}(p)$; if $l < \theta$ set $\theta \leftarrow l$, $i \leftarrow p$; and set $p \leftarrow \mathtt{RLINK}(p)$. (We could exit the loop immediately if $\theta = 0$.)

10. Item a is selected at level 0, trying option $x_0 = 12$, '$a\ d\ g$', and leading to (7). Then item b is selected at level 1, trying $x_1 = 16$, '$b\ c\ f$'. Hence, when the remaining item e is selected at level 2, it has no options in its list, and backtracking becomes necessary. Here are the current memory contents — substantially changed from Table 1:

i:	0	1	2	3	4	5	6	7
NAME(i):	—	a	b	c	d	e	f	g
LLINK(i):	0	0	0	0	3	0	5	6
RLINK(i):	0	2	3	5	5	0	0	0
x:	0	1	2	3	4	5	6	7
LEN(x):	—	2	1	1	1	0	0	1
ULINK(x):	—	20	16	9	27	5	6	25
DLINK(x):	—	12	16	9	27	5	6	25
x:	8	9	10	11	12	13	14	15
TOP(x):	0	3	5	−1	1	4	7	−2
ULINK(x):	—	3	5	9	1	4	7	12
DLINK(x):	10	3	5	14	20	21	25	18
x:	16	17	18	19	20	21	22	23
TOP(x):	2	3	6	−3	1	4	6	−4
ULINK(x):	2	9	6	16	12	4	18	20
DLINK(x):	2	3	6	22	1	27	6	25
x:	24	25	26	27	28	29	30	
TOP(x):	2	7	−5	4	5	7	−6	
ULINK(x):	16	7	24	4	10	25	27	
DLINK(x):	2	7	29	4	5	7	—	

11. Report that x is out of range if $x \le N$ or $x > Z$ or $\mathtt{TOP}(x) \le 0$. Otherwise set $q \leftarrow x$ and do "print $\mathtt{NAME(TOP}(q))$ and set $q \leftarrow q+1$; if $\mathtt{TOP}(q) \le 0$ set $q \leftarrow \mathtt{ULINK}(q)$" until $q = x$. Then set $i \leftarrow \mathtt{TOP}(x)$, $k \leftarrow 1$, and $q \leftarrow \mathtt{DLINK}(i)$. While $q \ne x$ and $q \ne i$, set $k \leftarrow k+1$ and $q \leftarrow$ `DLINK(q)`. If $q \neq i$, report that the option containing x is 'k of `LEN(i)`' in item i's list; otherwise report that it's not in that list.

12. For $0 \le j < l$, node x_j is part of an option in the solution. By setting $r \leftarrow x_j$ and then $r \leftarrow r+1$ until `TOP(r)` < 0, we'll know exactly what that option is: It's option number $-$`TOP(r)`, which begins at node `ULINK(r)`. (Many applications of Algorithm D have a custom-made output routine, to convert $x_0 \ldots x_{l-1}$ into an appropriate format — presenting it directly as a sudoku solution or a box packing, etc.)

Exercise 11 explains how to provide further information, not only identifying the option of x_j but also showing its position in the search tree.

13. In this case all of the uncovered item lists have the same length in step D3, so the choice if i doesn't affect the number of updates. If t items still haven't been covered, the list for item i consists of all 2^{t-1} subsets of those items that contain i; so we will do $\binom{t-1}{k-1}$ operations hide(p) on options p of size k. The total number of updates, u_t, when covering i in step D4 is therefore $1+\sum_k \binom{t-1}{k-1}(k-1) = 1+(t-1)2^{t-2}$.

Let x_n be the total number of updates to generate all ϖ_n of the set partitions. In step D5 we do $u_{n-1}+\cdots+u_{n-(k-1)}$ updates when the option containing x_l has k items. It follows that x_n satisfies the interesting recurrence

$$x_n \;=\; v_n + \sum_{k=1}^{n}\binom{n-1}{k-1}x_{n-k} \;=\; v_n + \sum_{k=0}^{n-1}\binom{n-1}{k}x_k, \qquad (*)$$

where $v_n = u_n + \sum_k \binom{n-1}{k-1}(u_{n-1}+\cdots+u_{n-(k-1)}) = ((9n-27)4^n - (8n-32)3^n + (36n-36)2^n + 72 - 41\delta_{n0})/72$. For example, $(x_0, x_1, \ldots, x_5) = (0, 1, 4, 18, 90, 484)$.

The general solution to $(*)$ is $x_n = \sum_{k=0}^{n} a_{nk}x_k$, where the matrix (a_{nk}) is

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 5 & 3 & 1 & 1 & 0 & 0 & 0 \\ 15 & 9 & 4 & 1 & 1 & 0 & 0 \\ 52 & 31 & 14 & 5 & 1 & 1 & 0 \\ 203 & 121 & 54 & 20 & 6 & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & -2 & -1 & 1 & 0 & 0 & 0 \\ -1 & -3 & -3 & -1 & 1 & 0 & 0 \\ -1 & -4 & -6 & -4 & -1 & 1 & 0 \\ -1 & -5 & -10 & -10 & -5 & -1 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}^{-1},$$

with rows and columns numbered from 0; for example, $x_4 = 15v_0 + 9v_1 + 4v_2 + v_3 + v_4$.

By Eq. 7.2.1.5–(14), $a_{n0} = \varpi_n$ is the total number of set partitions. And by the connection with Algorithm D, a_{nk} for $k > 0$ is the number of set partitions whose final block has size k. (For example, in 7.2.1.5–(2) the final block sizes are 4, 1, 1, 2, 1, 1, 2, 1, 2, 3, 1, 1, 1, 2, 1.) Computation shows that, as $n \to \infty$, the ratio a_{nk}/ϖ_n rapidly converges to a limiting value α_k, the probability that k is the size of the final block of a random set partition. We have, for instance, $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) \approx (.59635, .26597, .09678, .03009)$; and α_1 is the Euler–Gompertz constant, $-e\mathrm{Ei}(-1)$.

Computation also shows that these probabilities decrease so fast that we have

$$\lim_{n\to\infty} \frac{x_n}{\varpi_n} \;=\; 29.15616\;83478\;78328\;89209\;29419\;92431\;35251\;28901+;$$

thus, *the total number of updates per set partition is bounded, even though Algorithm D is working with exponentially many nodes*, and we've shown that the general-purpose dancing links approach is within a constant factor of Section 7.2.1.5's highly tuned methods. (Can this strange constant be expressed rigorously in closed form?)

sudoku
Bell number ϖ_n
final block
random set partition
Euler
Gompertz
closed form

14. Omit the options with $i = n - [n \text{ even}]$ and $j > n/2$.

(Other solutions are possible. For example, we could omit the options with $i = 1$ and $j \geq n$; that would omit $n-1$ options instead of only $\lfloor n/2 \rfloor$. However, the suggested rule turns out to make Algorithm D run about 10% faster.)

15. The two solutions are 'r_1 c_2 a_3 b_{-1}' 'r_2 c_4 a_6 b_{-2}' 'r_3 c_1 a_4 b_2' 'r_4 c_3 a_7 b_1' 'a_2' 'a_5' 'a_8' 'b_{-3}' 'b_0' 'b_3'; 'r_1 c_3 a_4 b_{-2}' 'r_2 c_1 a_3 b_1' 'r_3 c_4 a_7 b_{-1}' 'r_4 c_2 a_6 b_2' 'a_2' 'a_5' 'a_8' 'b_{-3}' 'b_0' 'b_3'. At the top levels, the MRV heuristic causes Algorithm D to branch first on the slack variables a_2, a_8, b_{-3}, and b_3, which each have at most two possibilities. (And that's actually a pretty strange way to tackle the queens' problem!)

16. Branch first on r_3, which has four options. If 'r_3 c_1 a_4 b_2', there's just one option for c_2, then c_3, then r_2, so we get the first solution: 'r_3 c_1 a_4 b_2' 'r_1 c_2 a_3 b_{-1}' 'r_4 c_3 a_7 b_1' 'r_2 c_4 a_6 b_{-2}'. If 'r_3 c_2 a_5 b_1', c_3 is forced, then r_2 can't be covered. If 'r_3 c_3 a_6 b_0', r_2 is forced, then c_2 can't be covered. If 'r_3 c_4 a_7 b_{-1}', we cruise to the second solution: 'r_3 c_4 a_7 b_{-1}' 'r_1 c_3 a_4 b_{-2}' 'r_2 c_1 a_3 b_1' 'r_4 c_2 a_6 b_2'. (And that's a good way.)

17. 'c e' 'a d f' 'b g' (as before) and 'b c f' 'a d g' (new).

18. When all primary items have been covered in step D2, accept a solution only if `LEN`$(i) = 0$ for all of the active secondary items, namely the items accessible from `RLINK`$(N+1)$. [This algorithm is called the "second death" method, because it checks that all of the purely secondary options have been killed off by primary covering.]

19. For $1 \leq k < m$, set $t \leftarrow k \mathbin{\&} (-k)$; include secondary item y_k in option α_j for $k \leq j < \min(m, k+t)$ and in option β_j for $k - t \leq j < k$.

Equivalently, to set up option α_j, include a and set $t \leftarrow j$; while $t > 0$, include y_t and set $t \leftarrow t \mathbin{\&} (t-1)$. To set up option β_j, include b and set $t \leftarrow -1-j$; while $t > -m$, include y_{-t} and set $t \leftarrow t \mathbin{\&} (t-1)$.

If $j > k$, options α_j and β_k both contain $y_{j \mathbin{\&} -2^{\lfloor \lg(k-j) \rceil}}$.

20. The options α_j^i will contain the primary item a_i. Simply do $k-1$ pairwise orderings, with secondary items y_k^i to ensure that $j_k \leq j_{k+1}$. If m is a power of 2, it turns out that the options for $1 < i < k$ each have exactly $\lg m$ secondary items. For example, if $m = 4$ and $k > 2$, the options α_j^2 are 'a_2 y_1^1 y_2^1', 'a_2 y_2^1 y_1^2', 'a_2 y_3^1 y_2^2', 'a_2 y_3^2 y_2^2'.

(The author attempted to knock out options for $\alpha^{i'}$ with $i' < i-1$ or $i' > i+1$, by adding additional secondary items, but that turned out to be a bad idea.)

Of course, this method doesn't compete with the lightning-quick methods for combination generation in Section 7.2.1.3; for instance, when $m = 20$ and $k = 8$ it needs 1.1 Gμ to crank out the $\binom{27}{8} = 2220075$ coverings, about 500 mems per solution.

21. (a) Let $n' = \lceil (n+1)/2 \rceil$. By rotation and reflection we can assume that the queen in column n' (the middle column) is in row i and the queen in row n' is in row j, where $1 \leq i < j < n'$. We obtain a suitable exact cover problem by leaving out the options $o(i,j) =$ 'r_i c_j a_{i+j} b_{i-j}' for $i = j$ or $i + j = n+1$; also omit $o(i,j)$ for $i > j$ when $j = n'$; $j > i$ when $i = n'$; and $(i,j) = (n'-1, n')$ or $(n', 1)$. Then include secondary items to force the pairwise ordering of $\alpha_k = o(k+1, n')$ and $\beta_k = o(n', k+2)$, for $0 \leq k < m = n'-2$.

(b) Now we assume a queen in (j,j), where $1 \leq j < n'$, and that the queen in row n is closer to the bottom right corner than the queen in column n. So we omit options $o(i,j)$ for $i+j = n+1$ or $i = j \geq n'$ or $(i,j) = (n,2)$ or $(i,j) = (n-1,n)$; we make item b_0 primary; and we let $\alpha_k = o(n, n-k-1)$, $\beta_k = o(n-k-2, n)$ for $0 \leq k < m = n-3$.

(c) This time we want queens in (i,i) and $(j, n+1-j)$ where $1 \leq i < j < n'$. We promote a_{n+1} and b_0 to primary; omit $o(i,j)$ when $i = j \geq n'-1$ or $i = n+1-j \geq n'$ or $(i,j) = (1,n)$; and let $\alpha_k = o(k+1, k+1)$, $\beta_k = o(k+2, n-k-1)$ for $0 \leq k < m = n'-2$.

MRV heuristic
second death
bitwise AND
author
combination generation

In case (a) there are (0, 0, 1, 8, 260, 9709, 371590) solutions for $n = (5, 7, \ldots, 17)$; Algorithm D handles $n = 17$ in 3.4 Gμ. [In case (b) there are (0, 0, 1, 4, 14, 21, 109, 500, 2453, 14498, 89639, 568849) for $n = (5, 6, \ldots, 16)$; and $n = 16$ costs 6.0 Gμ. In case (c), similarly, there are (1, 0, 3, 6, 24, 68, 191, 1180, 5944, 29761, 171778, 1220908) solutions; $n = 16$ costs 5.5 Gμ.]

pairwise comparisons
author
asymmetric
singly symmetric
doubly symmetric

22. (a) Consider the queens in column a of row 1, row b of column n, column $\bar{c}$ of row n, and row $\bar{d}$ of column 1, where $\bar{x} = n + 1 - x$. (These four queens are distinct, because no queen is in a corner. Notice also that neither $\bar{a}$ nor $\bar{b}$ nor $\bar{c}$ nor $\bar{d}$ can equal a.) Repeated rotations and/or reflections will change these numbers from (a, b, c, d) to

$$(b, c, d, a),\ (c, d, a, b),\ (d, a, b, c),\ (\bar{d}, \bar{c}, \bar{b}, \bar{a}),\ (\bar{c}, \bar{b}, \bar{a}, \bar{d}),\ (\bar{b}, \bar{a}, \bar{d}, \bar{c}),\ (\bar{a}, \bar{d}, \bar{c}, \bar{b}).$$

Those eight 4-tuples are usually distinct, and in such cases we can save a factor of 8 by eliminating all but one of them. There always is a solution with $a \le b, c, d < \bar{a}$; and those inequalities can be enforced by doing three simultaneous pairwise comparisons, between the options for row 1 and the respective options for column n, row n, and column 1. For example, the options that correspond to $a = 1$ when $n = 16$ are 'r_1 c_2 a_3 b_{-1}'; 'r_2 c_{16} a_{18} b_{-14} x_1 x_2 x_4'; 'r_{15} c_{16} a_{31} b_{-1} x_1 x_2 x_4'; 'r_{16} c_2 a_{18} b_{14} y_1 y_2 y_4'; 'r_{16} c_{14} a_{30} b_2 y_1 y_2 y_4'; 'r_2 c_1 a_3 b_1 z_1 z_2 z_4'; 'r_{15} c_1 a_{16} b_{14} z_1 z_2 z_4'. (Here $m = n/2 - 1 = 7$.)

With this change, the number of solutions for $n = 16$ drops from 454376 to 64374 (ratio ≈ 7.06), and the running time drops from 4.3 Gμ to 1.2 Gμ (ratio ≈ 3.68).

[The author experimented with further restrictions, so that solutions were allowed only if (i) $a < b, c, d$; (ii) $a = b < c, d$; (iii) $a = b = c < d$; (iv) $a = b = c = d$; (v) $a = c < b, d$. Five options were given for each value of $a < n/2 - 1$, and m was 6 instead of 7. The number of solutions decreased to 59648; but the running time increased to 1.9 Gμ. Thus a point of diminishing returns had been reached. (A completely canonical reduction would have produced 57188 solutions, with considerable difficulty.)]

(b) This case is almost identical to (a), because the queen in the center vacates all other diagonal cells. Requiring $a \le b, c, d < \bar{a}$ reduces the number of solutions for $n = 17$ from 4067152 to 577732 (ratio ≈ 7.04), and run time to 3.2 Gμ (ratio ≈ 4.50).

23. We simply combine compatible options into (a) pairs, (b) quadruplets, and force a queen in the center when n is odd. For example, when $n = 4$ we replace (23) by (a) 'r_1 c_2 a_3 b_{-1} r_4 c_3 a_7 b_1'; 'r_1 c_3 a_4 b_{-2} r_4 c_2 a_6 b_2'; 'r_2 c_1 a_3 b_1 r_3 c_4 a_7 b_{-1}'; 'r_2 c_4 a_6 b_{-2} r_3 c_1 a_4 b_2'; (b) 'r_1 c_2 a_3 b_{-1} r_2 c_4 a_6 b_{-2} r_4 c_3 a_7 b_1 r_3 c_1 a_4 b_2'; 'r_2 c_1 a_3 b_1 r_3 c_4 a_7 b_{-1} r_1 c_3 a_4 b_{-2} r_4 c_2 a_6 b_2'. The options when $n = 5$ are (a) 'r_1 c_2 a_3 b_{-1} r_5 c_4 a_9 b_1'; 'r_1 c_4 a_5 b_{-3} r_5 c_2 a_7 b_3'; 'r_2 c_1 a_3 b_1 r_4 c_5 a_9 b_{-1}'; 'r_2 c_5 a_7 b_{-3} r_4 c_1 a_5 b_3'; 'r_3 c_3 a_6 b_0'; (b) 'r_1 c_2 r_2 c_5 r_5 c_4 r_4 c_1'; 'r_2 c_1 r_1 c_4 r_4 c_5 r_5 c_2'; 'r_3 c_3 a_6 b_0'.

An n-queen solution is either *asymmetric* (changed by 180° rotation) or *singly symmetric* (changed by 90° rotation but not 180°) or *doubly symmetric* (unchanged by 90° rotation). Let $Q_{\rm a}(n)$, $Q_{\rm s}(n)$, $Q_{\rm d}(n)$ be the number of such solutions that are essentially different; then $Q(n) = 8Q_{\rm a}(n) + 4Q_{\rm s}(n) + 2Q_{\rm d}(n)$ when $n > 1$. Furthermore there are $4Q_{\rm s}(n) + 2Q_{\rm d}(n)$ solutions to (a) and $2Q_{\rm d}(n)$ solutions to (b). Hence we can determine the individual values just by counting solutions, and we obtain these results for small n:

$n =$	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$Q_{\rm a}(n) =$	0	1	0	4	11	42	89	329	1765	9197	45647	284743	1846189	11975869
$Q_{\rm s}(n) =$	0	0	1	2	1	4	3	12	18	32	105	310	734	2006
$Q_{\rm d}(n) =$	1	1	0	0	0	0	0	0	4	4	0	0	32	64

We can reduce the solutions to (a) by a factor of 2, by simply eliminating the options that contain $\{r_1, c_k\}$ for $k \ge \lceil n/2 \rceil$. We can reduce the solutions to (b) by a factor of $2^{\lfloor n/4 \rfloor}$, by simply eliminating the options that contain $\{r_j, c_k\}$ for $j < \lceil n/2 \rceil$ and $k \geq \lceil n/2 \rceil$. With these simplifications, the computation of $Q_{\rm d}(16)$ needs only 70 Kμ; and then the computation of $Q_{\rm s}(16)$ needs only 5 Mμ. Only 20 Mμ are needed to determine that $Q_{\rm d}(32) = 2^7 \cdot 1589$.

eight queens problem
Gosset
Dudeney
gadget

24. With 64 items, one for each cell of the chessboard, let there be 92 options, one for each of the 92 solutions to the eight queens problem (see Fig. 68). Every option names eight of the 64 items; so an 8-coloring is equivalent to solving this exact cover problem. Algorithm D needs only 25 kilomems and a 7-node search tree to show that such a mission is impossible. [In fact no *seven* solutions can be disjoint, because each solution touches at least three of the twenty cells 13, 14, 15, 16, 22, 27, 31, 38, 41, 48, 51, 58, 61, 68, 72, 77, 83, 84, 85, 86. See Thorold Gosset, *Messenger of Mathematics* **44** (1914), 48. However, Henry E. Dudeney found the illustrated way to occupy all but two cells, in *Tit-Bits* **32** (11 September 1897), 439; **33** (2 October 1897), 3.]

```
12345678
78563412
46718235
23854167
84236751
51672384
67481523
 512784
```

25. This is an exact cover problem with $92 + 312 + 396 + \cdots + 312 = 3284$ options (see exercise 7.2.2–6). Algorithm D needs about 32 megamems to find the solution shown, and about 1.3 Tμ to find all 11,092 of them.

```
07348652
18650437
75421860
26835071
34072186
52183704
80564213
61207345
```

26. Let u_{jh} and d_{jh} be secondary items for $1 \leq j \leq 2n$ and $1 \leq h \leq \lceil n/2 \rceil$. Insert the gadget

$$u_{j1}\ u_{j2}\ \ldots\ u_{j\lceil i/2\rceil}\ u_{(j+1)\lceil i/2\rceil}\ \ldots\ u_{k\lceil i/2\rceil}\ \ldots\ u_{k2}\ u_{k1}$$

into each option (16); also append similar options, but with 'u' changed to 'd', except when $i = n$. [Solutions whose planar graph "splits" will be obtained more than once. One such example is 12 10 8 6 4 11 9 7 5 4 6 10 12 5 7 9 11 3 1 2 1 3 2.]

27. (a) Denoting that formula by $\rho(c_0, t_0; \ldots; c_l, t_l)$, notice that if $c'_j = t_j + 1 - c_j$ we have $\rho(c_0, t_0; \ldots, c_l, t_l) + \rho(c'_0, t_0; \ldots, c'_l, t_l) = 1$. Consequently the completion ratio is $1/2$ if and only if $c'_j = c_j$ for all j, namely when $t_j = 2c_j - 1$.

(b) The ratio $\rho(c_0, t_0; \ldots; c_l, t_l)$ never has an odd denominator, because $p/q + p'/q'$ has an even denominator whenever q and p' are odd and q' is even. But we can get arbitrarily close to $1/3$, since $\rho(2, 4; \ldots; 2, 4) = 1/3 + 1/(24 \cdot 4^l)$.

28. If T has only a root node, let there be one column, no rows. Otherwise let T have $d \geq 1$ subtrees $T_1, \ldots, T_d$, and assume that we've constructed matrices with rows R_j and columns C_j for each T_j. Let $C = C_1 \cup \cdots \cup C_d \cup \{1, \ldots, d\}$. The matrix for T is obtained by appending $d{+}1$ new columns $\{0, 1, \ldots, d\}$ and the following new rows: (i) '0 and all columns of $C \setminus C_j$', for $1 \leq j \leq d$; (ii) 'all columns of $C \setminus j$', for $1 \leq j \leq d$. This construction works except when $d = 1$ and T_1 is a leaf; in that case we can use the 3×4 matrix whose rows are 1110, 0101, 0011. The matrix for the example tree has 17 columns and 16 rows.

```
01111110000000000
10111110000000000
11011110000000000
11100110000000000
11101010000000000
11100110000000000
00000000111111000
00000001011111000
00000001101111000
00000001110011000
00000001110101000
00000001110110000
00000001111111111
11111110000000111
11111111111111001
11111111111111010
```

29. (a) If a solution isn't at the root, its parent must have exactly one child. (Alternatively, if duplicate options are permitted, all siblings of a solution must be solutions.)

(b) Use the previous construction; a solution node corresponds to the matrix '1'.

31. (a) No. Otherwise there would be an option with no primary items.

(b) Yes, but only if there are two options with the same primary items.

(c) Yes, but only if there are two options whose union is also an option, when restricted to primary items.

(d) The number of places, j, where $x = 1$ and $x' = 0$ must be the same as the number where $x = 0$ and $x' = 1$. For if A has exactly k primary items in every option, exactly jk primary items are being covered in different ways.

(e) Again distances must be even, because every solution also solves the restricted problem, which is uniform. (Consequently it makes sense to speak of the *semidistance* $d(x, x')/2$ between solutions of a quasi-uniform exact covering problem. The semidistance in a polyform packing problem is the number of pieces that are packed differently.)

33. (Solution by T. Matsui.) Add one new column at the left of A, all 0s. Then add two rows of length $n + 1$ at the bottom: $10\ldots0$ and $11\ldots1$. This $(m + 2) \times (n + 1)$ matrix A' has one solution that chooses only the last row. All other solutions choose the second-to-last row, together with rows that solve A.

34. (Solution by T. Matsui.) Assume that all 1s in column 1 appear in the first t rows, where $t > 3$. Add two new columns at the left, and two new rows $1100\ldots0$, $1010\ldots0$ of length $n + 2$ at the bottom. For $1 \le k \le t$, if row k was $1\alpha_k$, replace it by $010\alpha_k$ if $k \le t/2$, $011\alpha_k$ if $k > t/2$. Insert 00 at the left of the remaining rows $t + 1$ through m.

This construction can be repeated (with suitable row and column permutations) until no column sum exceeds 3. If the original column sums were $(c_1, \ldots, c_n)$, the new A' has $2T$ more rows and $2T$ more columns than A did, where $T = \sum_{j=1}^{n}(c_j \dot- 3)$.

One consequence is that the exact cover problem is NP-complete even when restricted to cases where all row and column sums are at most 3.

Notice, however, that this construction is *not* useful in practice, because it disguises the structure of A: It essentially *destroys* the minimum remaining values heuristic, because all columns whose sum is 2 look equally good to the solver!

35. Take a matrix with column sums $(c_1, \ldots, c_n)$, all ≤ 3, and extend it with three columns of 0s at the right. Then add the following four rows: $(x_1, \ldots, x_n, 0, 1, 1)$, $(y_1, \ldots, y_n, 1, 0, 1)$, $(z_1, \ldots, z_n, 1, 1, 0)$, and $(0, \ldots, 0, 1, 1, 1)$, where $x_j = [c_j < 3]$, $y_j = [c_j < 2]$, $z_j = [c_j < 1]$. The bottom row must be chosen in any solution.

36. Consider a set of cubes and colors called $\{*, 0, 1, 2, 3, 4, \ldots\}$, where (i) all faces of cube $*$ are colored $*$; (ii) colors 1, 2, 3, 4 occur only on cubes 0, 1, 2, 3, 4; (iii) the opposite face-pairs of those five cubes are respectively $(00, 12, **)$, $(11, 12, 34)$, $(22, 34, \alpha)$, $(33, 12, \beta)$, $(44, 34, \gamma)$, where α, β, γ are pairs of colors $\notin \{1, 2, 3, 4\}$. Any solution to the cube problem has disjoint 2-regular graphs X and Y containing two faces of each color. Since X and Y both contain $**$ from cube $*$, we can assume that X contains 00 and Y contains 12 from cube 0. Hence Y can't contain 11 or 22; it must contain 12 from cube 1 or cube 3. If X doesn't contain 11 or 22, it must contain 12 from cube 1 *and* cube 3. Hence X contains 11, 22, 33, and 44. We're left with only three possibilities for Y from cubes 1, 2, 3, 4, namely $(34, \alpha, 12, 34)$, $(12, 34, \beta, 34)$, $(34, 34, 12, \gamma)$.

Now let a_{j1}, a_{j2}, a_{j3} denote the 1s in column j of A. We construct $N = 8n + 1$ cubes and colors called $*$, a_{jk}, b_{jl}, where $1 \le j \le n$, $1 \le k \le 3$, $0 \le l \le 4$. The opposite face-pairs of $*$ are $(**, **, **)$. Those of a_{jk} are $(a_{jk}a_{jk}, a_{jk}a_{jk}, a_{jk}b_{j'0})$, where j' is the column of a_{jk}'s cyclic successor to the right in its row. Those of b_{j0}, b_{j1}, b_{j2}, b_{j3}, b_{j4} are respectively $(b_{j0}b_{j0}, b_{j1}b_{j2}, **)$, $(b_{j1}b_{j1}, b_{j1}b_{j2}, b_{j3}b_{j4})$, $(b_{j2}b_{j2}, b_{j3}b_{j4}, b_{j0}a_{j1})$, $(b_{j3}b_{j3}, b_{j1}b_{j2}, b_{j0}a_{j2})$, $(b_{j4}b_{j4}, b_{j3}b_{j4}, b_{j0}a_{j3})$. By the previous paragraph, solutions to the cube problem correspond to 2-regular graphs X and Y such that, for each j, X or Y contains all the pairs $b_{jl}b_{jl}$ and the other "selects" one of the three pairs $b_{j0}a_{jk}$. The face-pairs of each selected a_{jk} ensure that a_{jk}'s cyclic successor is also selected.

[See E. Robertson and I. Munro, *Utilitas Mathematica* **13** (1978), 99–116.]

semidistance
Matsui
Matsui
NP-complete
minimum remaining values heuristic
2-regular graphs
Robertson
Munro

50. Set $f_m \leftarrow 0$ and $f_{k-1} \leftarrow f_k \mid r_k$ for $m \ge k > 1$. The bits of u_k represent items that are being changed for the last time.

Let $u_k = u' + u''$, where $u' = u_k \,\&\, p$. If $u_k \ne 0$ at the beginning of step N4, we compress the database as follows: For $N \ge j \ge 1$, if $s_j \,\&\, u' \ne u'$, delete (s_j, c_j); otherwise if $s_j \,\&\, u'' \ne 0$, delete (s_j, c_j) and insert $((s_j \,\&\, \bar{u}_k) \mid u', c_j)$.

To delete (s_j, c_j), set $(s_j, c_j) \leftarrow (s_N, c_N)$ and $N \leftarrow N - 1$.

When this improved algorithm terminates in step N2, we always have $N \le 1$. Furthermore, if we let $p_k = r_1 \mid \cdots \mid r_{k-1}$, the size of N never exceeds 2^{ν_k}, where $\nu_k = \nu\langle p_k r_k f_k\rangle$ is the size of the "frontier" (see exercise 7.1.4–55).

[In the special case of n queens, represented as an exact cover problem as in (23), this algorithm is due to I. Rivin, R. Zabih, and J. Lamping, *Inf. Proc. Letters* **41** (1992), 253–256. They proved that the frontier for n queens never has more than $3n$ items.]

51. The author has had reasonably good results using a triply linked binary search tree for the database, with randomized search keys. (Beware: The swapping algorithm used for deletion was difficult to get right.) This implementation was, however, limited to exact cover problems whose matrix has at most 64 columns; hence it could do n queens via (23) only when $n < 12$. When $n = 11$ its database reached a maximum size of 75,009, and its running time was about 25 megamems. But Algorithm D was noticeably better: It needed only about 12.5 Mμ to find all $Q(11) = 2680$ solutions.

In theory, this method will need only about 2^{3n} steps as $n \to \infty$, times a small polynomial function of n. A backtracking algorithm such as Algorithm D, which enumerates each solution explicitly, will probably run asymptotically slower (see exercise 7.2.2–15). But in practice, a breadth-first approach needs too much space.

On the other hand, this method did beat Algorithm D on the n queen bees problem of exercise 7.2.2–16: When $n = 11$ its database grew to 364,864 entries; it computed $H(11) = 596{,}483$ in just 30 Mμ, while Algorithm D needed 440 Mμ.

52. The set of solutions for s_j can be represented as a regular expression α_j instead of by its size, c_j. Instead of inserting $(s_j + t, c_j)$ in step N3, insert $\alpha_j k$. If inserting (s, α), when (s_i, α_i) is already present with $s_i = s$, change $\alpha_i \leftarrow \alpha_i \cup \alpha$. [Alternatively, if only one solution is desired, we could attach a single solution to each s_j in the database.]

59. Let $i = (i_1 i_0)_3$ and $j = (j_1 j_0)_3$; then cell (i, j) belongs to box $(i_1 j_1)_3$. Mathematically, it's better to consider the matrices $a'_{ij} = a_{ij} - 1$, $b'_{ij} = b_{ij} - 1$, $c'_{ij} = c_{ij} - 1$, which are the "multiplication tables" of interesting binary operators on $\{0, \ldots, 8\}$. We have $a'_{ij} = ((i_0 i_1)_3 + j) \bmod 9$; $b'_{ij} = ((i_0 + j_1) \bmod 3, (i_1 + j_0) \bmod 3)_3$; and $c'_{ij} = ((i_0 + i_1 + j_1) \bmod 3, (i_0 - i_1 + j_0) \bmod 3)_3$. (Furthermore the latter two operators are "isotopic": $c'_{ij} = b'_{(i\pi)(j\pi^{-1})}\pi$, when $(i_1, i_0)_3\pi = (i_1, (i_0 + i_1) \bmod 3)_3$.)

[A pattern like (28c) appeared in a Paris newspaper of 1895, in connection with magic squares. But no properties of its 3×3 subsquares were mentioned; it was a sudoku solution purely by coincidence. See C. Boyer, *Math. Intelligencer* **29**, 2 (2007), 63.]

60. No. The 33rd digit is 0.

61. Step D3 chooses p_{44}, p_{84}, p_{74}, p_{24}, p_{54}, p_{14}, p_{82}, p_{42}, p_{31}, p_{32}, p_{40}, p_{45}, p_{46}, p_{50}, p_{72}, p_{60}, p_{00}, p_{62}, p_{61}, p_{65}, p_{35}, p_{67}, p_{70}, p_{71}, p_{75}, p_{83}, p_{13}, p_{03}, p_{18}, p_{16}, p_{07}, p_{01}, p_{05}, p_{15}, p_{21}, p_{25}, p_{76}, p_{36}, p_{33}, p_{37}, p_{27}, p_{28}, p_{53}, p_{56}, p_{06}, p_{08}, p_{58}, p_{77}, p_{88}, in that order.

62. The lists for items p_{44}, p_{84}, r_{33}, r_{44}, r_{48}, r_{52}, r_{59}, r_{86}, r_{88}, c_{22}, c_{43}, b_{07}, b_{32}, b_{39}, b_{43}, b_{54}, and b_{58} have length 1 when Algorithm D begins to tackle puzzle (29a). Step D3 will branch on whichever item was placed first in step D1. (The author's sudoku setup program puts p before r before c before b in that step.)

frontier
n queens
Rivin
Zabih
Lamping
author
triply linked
binary search tree
backtracking algorithm
asymptotically
theory vs practice
practice vs theory
n queen bees
regular expression
0-origin indexing
multiplication tables
binary operators
isotopic
magic squares
sudoku solution
Boyer
author
sudoku setup program

63. r_{13}, c_{03}, b_{03}, b_{24}, b_{49}, b_{69}. The latter three were hidden already in (32).

64. In case (a) we list the available columns; in case (b) we list the available rows:

(a)

	1	2	3	4	5	6	7	8	9
0	45	**3**	**6**	**8**	**0**	12457	1245	2457	157
1	**0**	278	**1**	**3**	**5**	2478	2468	2467	78
2	**6**	0127	**3**	012	**8**	012457	01245	02457	157
3	1234578	1278	78	12456	27	12345	1234568	24567	**0**
4	12378	**5**	**0**	12	27	**6**	1238	27	**4**
5	12378	01278	**5**	0126	**6**	0123	012368	0267	78
6	4578	**6**	78	045	**1**	04578	045	**3**	**2**
7	125	012	**4**	0125	**6**	0125	**7**	**8**	**3**
8	58	**4**	**2**	**7**	**3**	058	05	**1**	**6**

(b)

	0	1	2	3	4	5	6	7	8
1	**1**	3457	3457	345	036	03678	**2**	3456	34568
2	257	2357	12357	**0**	**8**	**4**	**6**	1235	135
3	**4**	**1**	**8**	**2**	**7**	**5**	**0**	36	36
4	2567	23457	23457	**1**	36	367	35	**8**	**0**
5	**0**	**6**	34	**8**	**5**	**1**	**7**	34	**2**
6	25678	02357	012357	35	01236	023678	**4**	0126	168
7	2568	02345	012345	345	01236	02368	135	**7**	1345
8	25	**8**	012345	**6**	0123	023	135	012345	**7**
9	**3**	02	**6**	**7**	**4**	02	**8**	0125	15

(Notice that "hidden" singles and pairs, etc., become "naked" in this representation. Similar plots, which relate boxes to values, are also possible; but they're trickier, because boxes aren't orthogonal to rows or columns.)

65. (a) For columns, remove all items r_{ik} and b_{xk}, as well as c_{jk} with $j \neq j_0$; let u_j —— v_k when an option contains '$p_{ij_0}\ c_{j_0k}$'. For boxes, remove all r_{ik}, c_{jk}, and b_{xk} with $x \neq x_0$; let u_j —— v_k when an option contains '$p_{(3\lfloor x_0/3\rfloor+\lfloor j/3\rfloor)(3(x \bmod 3)+(j \bmod 3))}\ b_{x_0k}$'.

(b) The $n-q$ *non*-neighbors of a hidden q-tuple (e.g., $\{u_3, u_8, u_1\}$) are "naked."

(c) By (b) it suffices to list the naked ones (and only those for which $q < r$). Let's denote the option in (30) by ijk. In row 4 we find the naked pair $\{u_3, u_8\}$, hence can delete options 411, 417, 421, 427, 471; also the naked triple $\{u_1, u_3, u_8\}$, so can also delete option 424. There's no nakedness in the columns. The naked triple $\{u_0u_3u_6\}$ in box 4 allows deletion of options 341, 346, 347, 351, 356, 357.

(d) Let u_i —— v_j if there's an option that contains '$r_{ik_0}\ c_{jk_0}$'. When $k_0 = 9$ there's a naked pair $\{u_1, u_5\}$, so we can delete options 079 and 279.

[Many other reductions have been proposed. For example, (33) has a "pointing pair" in box 4: Since '4' and '8' must occupy that box in row 3, we can remove options 314, 324, 328, 364, 368, 378. Classic references are the early tutorials by W. Gould, *The Times Su Doku Book 1* (2005); M. Mepham, *Solving Sudoku* (2005).]

66. Such a puzzle must add a 7 or 8 in one of 18 places, because (29c) has just 2 solutions. So there are 36 of them (18 isomorphic pairs).

67. We can solve this problem with Algorithm M, using options (30) with $k \neq 8$ and giving multiplicity 2 to each of the items r_{i7}, c_{j7}, b_{x7}. There are six solutions, all of which extend the partial solution shown. Only one yields a sudoku square when we change half of the 7s to 8s.

9	3	4	5	1	7		6	
7	6	2	4	9	3	1	7	5
7	5	1	7			4	9	3
2	7	5	9	7	1	6	3	4
6	4	9		3	5		1	
1	7	3				5		9
4	1	7	6	5	9	3		
3	2	7	1			9	5	6
5	9	6	3			7	4	1

68. For example, removing clues one at a time shows that only 10 of the 32 givens are actually essential. The best strategy for finding all minimal X is probably to examine candidate sets in order of decreasing cardinality: Suppose $W \subseteq X$, and suppose that previous tests have shown that the solution is unique, given X, but not given $X \setminus w$ for any $w \in W$. We have a solution if $W = X$. Otherwise let $X \setminus W = \{x_1, \ldots, x_t\}$, and test $X \setminus x_i$ for each i. Suppose the solution turns out to be unique if and only if $i > p$. Then we schedule the $t-p$ candidate pairs $(W \cup \{x_1, \ldots, x_p\}, X \setminus x_i)$, $p < i \leq t$, for processing in the next round. With suitable caching of previous results, we can avoid testing the

hidden
naked
pointing pair
Gould
Mepham

same subset of clues more than once. Furthermore we can readily modify Algorithm D so that it backtracks immediately after discovering a single unwanted solution.

All 777 minimal subsets were found in this manner, involving 15441 invocations of Algorithm D, but needing a total of only about 1.5 gigamems of computation. Altogether (1, 22, 200, 978, 2780, 4609, 4249, 1950, 373, 22) candidate pairs were examined in rounds (32, 31, …, 23); and exactly (8, 154, 387, 206, 22) solutions were found of sizes (27, 26, 25, 24, 23). The lexicographically last 23-clue subset, which is illustrated below, turns out to be a fairly tough puzzle, with 219 nodes in its search tree.

(Let $f(x_1, \ldots, x_{32})$ be the monotone Boolean function '[the solution is unique, given the clues with $x_j = 1$]'. This problem essentially asks for f's prime implicants.)

(29a)

				1				
4	1						9	
		6	5					
5				8				9
	7						3	2
	3	8			4			
			2	6		4		
			3					8
3					7		5	

; (28a)

1	2	3						
			7	8	9			
2	3	4						
						5	6	7
		5	6	7				
9	1							8

; (28b)

1					6		8	
	5					1		
		9				4		
	3		5					7
				9				
8					1			
							7	
6	4				8			2
			3					5

.

69. If only one of those nine appearances has been specified, the other eight can always be permuted into another solution. And the entire diagram can be partitioned into nine disjoint sets of nine, all with the same property, thus requiring at least $2 \cdot 9$ clues.

This argument proves that all 18-clue characterizations must have a very special form. The interesting solution above makes a particularly satisfying puzzle. (The author found it with the help of a SAT solver; see Section 7.2.2.2.)

The same argument shows that (28b) needs at least 18 clues. But this time the corresponding SAT instance is unsatisfiable. Moreover, any 19-clue solution must have three clues in just one critical group of nine; the associated SAT instance, which insists on having at least one clue in each of the 2043 subsets of at most 18 cells that can be rearranged into new solutions, also is unsatisfiable. (Proved in 177 Mμ.) But hurray, the special structure does lead to 20-clue examples, like the one above.

(The constructions for (28b) apply also to (28c), via the isotopism in answer 59.)

70. When ps precede rs precede cs precede bs in D1, the tree sizes are 1105, 910, 122.

(34a)

3	1	2	5	8	6	7	4	9
9	4	1	8	3	7	2	5	6
2	6	5	9	4	8	3	1	7
7	3	4	2	6	9	1	8	5
1	8	9	7	5	3	6	2	4
6	9	7	4	2	5	8	3	1
5	2	3	6	1	4	9	7	8
8	5	6	1	7	2	4	9	3
4	7	8	3	9	1	5	6	2

; (34b)

G	T	A	R	D	N	E	I	M
D	N	T	G	I	R	A	M	E
I	G	E	T	N	A	M	D	R
T	I	R	N	A	M	D	E	G
R	A	I	D	M	E	N	G	T
M	R	N	I	E	D	G	T	A
A	M	D	E	R	G	T	N	I
N	E	M	A	G	T	I	R	D
E	D	G	M	T	I	R	A	N

; (34c)

W	V	Y	N	A	L	T	K	F
F	T	L	W	K	Y	A	N	V
K	A	V	T	N	F	L	W	Y
N	K	F	L	T	V	W	Y	A
Y	F	T	A	L	W	N	V	K
A	L	W	K	V	N	Y	F	T
V	W	N	Y	F	A	K	T	L
L	Y	K	F	W	T	V	A	N
T	N	A	V	Y	K	F	L	W

.

71. Using the options (30), items r_{ik} and c_{jk} should be *secondary* when row i or column j contains fewer than 6 cells. The puzzles are fun to solve by hand; but in a pinch, Algorithm D will traverse search trees of sizes 23, 26, and 16 to find the answers:

(a)

	1	5	6	2	
5	4	6	1	3	2
6	2			4	3
1	5	3	2	6	4
2	6	4	3	5	1
4	3			1	5

; (b)

3	2	1	5	6	
6	5		4	1	3
4	1	5	6	2	
1	4	3	2	5	6
5	3			4	2
2	6	4	1	3	

; (c)

	3	6	4	5	
4	5	3	6	2	1
2	1				
1	2				
3	4	2	1	6	5
	6	5	3	4	

.

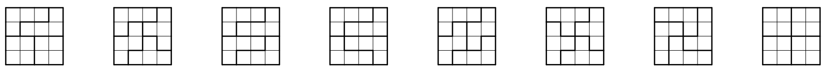
monotone Boolean function
order ideals
prime implicants
author
SAT solver
isotopism
secondary

[These are the first of 26 elegant puzzles announced by Serhiy and Peter Grabarchuk on Martin Gardner's 100th birthday (21 October 2014) and posted at `puzzlium.com`.]

72. Exactly 1315 of the $\binom{25}{5} = 53130$ ways to retain five clues result in a unique solution, and 175 of them involve all five digits. The lexicographically first is Fig. A–2(a).

73. Follow the hint; the undesired *straight* n-ominos can be rejected easily in step R2 by examining v_{n-1} and v_0. This quickly produces (16, 105, 561, 2804, 13602) box options, for $n = (3, 4, 5, 6, 7)$, which can be fed to Algorithm D to get jigsaw patterns.

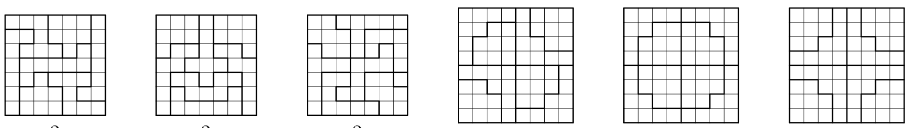
There are no patterns for $n = 3$. But $n = 4$ has 33 patterns, which divide into eight equivalence classes under rotation and/or reflection:



1 1 2 2 2 4 4 8

(The number of symmetries is shown below each arrangement; notice that $8/1 + 8/1 + 8/2 + 8/2 + 8/2 + 8/4 + 8/4 + 8/8 = 33$.) Similarly, $n = 5$ has 266 equivalence classes, representing $256 \cdot (8/1) + 7 \cdot (8/2) + 3 \cdot (8/4) = 2082$ total patterns; $n = 6$ has 40237 classes, representing $39791 \cdot (8/1) + 439 \cdot (8/2) + 7 \cdot (8/4) = 320098$ patterns in all.

The computation gets more serious in the case $n = 7$, when Algorithm D needs about 1.9 Tμ to generate the 132,418,528 jigsaw patterns. These patterns include 16,550,986 classes with no symmetry, and 2660 with one nontrivial symmetry. The latter break down into 2265 that are symmetric under 180° rotation, 354 that are symmetric under horizontal reflection, and 41 that are symmetric under diagonal reflection. Here are some typical symmetric examples:



2 2 2 4 8 8

(It's not difficult to generate all of the *symmetric* solutions for slightly higher values of n; three of the classes for $n = 8$, shown above, have more than 2 symmetries. And the case $n = 9$ contains two patterns with 8-fold symmetry *besides* the standard sudoku boxes: See Fig. A–2(b) and (c), where the latter might be called windmill sudoku! For complete counts for $n = 8$ and $n = 9$, with straight n-ominoes allowed, see Bob Harris's preprint "Counting nonomino tilings," presented at G4G9 in 2010.)

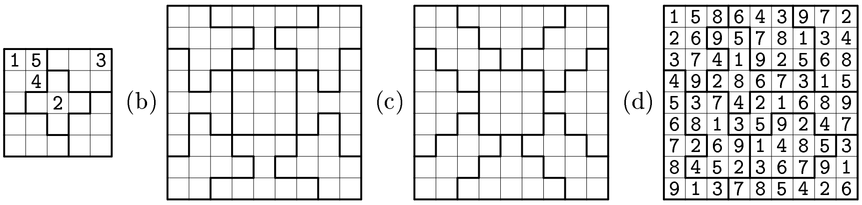


Fig. A–2. Jigsaw sudoku patterns.

74. A simple modification of exercise 7.2.2–101 will generate the 3173 boxes that have the desired rainbow property. An exact cover problem, given those 3173 options, shows (after 1.2 Gμ of computation) that the boxes can be packed in 98556 ways. If we restrict the options to the 3164 that aren't sudoku boxes, the number of packings goes down to 42669, of which 24533 are faultfree. Figure A–2(d) is a faultfree example.

S Grabarchuk
P Grabarchuk
Gardner
straight n-ominos
rotation
reflection
symmetries
180° rotation
circular symmetry
reflection
reflection symmetry
8-fold symmetry
windmill sudoku
Harris
faultfree

Harris
diagonal
author
Henle
Brouwer
isomorphic
isomorphic

75. (a) When $n = 4$, one of the eight classes in answer 73 (the 2nd) has *no* solutions; another (the 5th) is clueless. When $n = 5$, eight of the 266 classes have no solution; six are clueless. When $n = 6$, 1966 of 40237 are vacuous and 28 are clueless.

(Maxime's original puzzle appeared in the newsletter of Chicago Area Mensa [*ChiMe* **MM**, 3 (March 2000), 15]. Algorithm D solves it with a 39-node search tree. But the tree size would have been 214 if he'd put `ABCDEF` in the next row down!)

(b) (Solution by Bob Harris, `www.bumblebeagle.org/dusumoh/proof/`, 2006.) The clueless jigsaw for $n = 4$ generalizes to all larger n, as illustrated here for $n = 7$: First $a = 3$; hence $b = 3$; ...; hence $f = 3$. Then $g = 4$; hence $h = 4$; ...; hence $l = 4$. And so on. Finally we know where to place the 2's and the 1's. (This proof shows that, for odd $n > 3$, there's always an $n \times n$ jigsaw sudoku whose clues lie entirely on the main diagonal. Is there also a general construction that works for *even* values of n? An 8×8 example appears in exercise 76.)

1	2	3	4	5	6	7
			a	g		
				b	h	
					c	i
k						d
f	l					
	e	j				

76. (The author designed these puzzles with the aid of exercises 73 and 75. Similar puzzles have been contrived by J. Henle, *Math. Intelligencer* **38**, 1 (2016), 76–77.)

D	G	C	I	A	·	N
N	A	I	G	·	D	C
G	D	N	A	C	I	·
I	·	A	C	N	G	D
·	N	G	D	I	C	A
A	C	D	·	G	N	I
C	I	·	N	D	A	G

L	K	S	N	I
N	I	K	S	L
I	S	N	L	K
S	L	I	K	N
K	N	L	I	S

C	P	T	M	R	E	O	U
U	O	P	R	E	C	T	M
E	C	M	U	O	R	P	T
R	T	U	P	M	O	C	E
O	R	E	T	U	P	M	C
M	U	C	E	P	T	R	O
T	M	R	O	C	U	E	P
P	E	O	C	T	M	U	R

A	I	L	T	G	H	O	M	R
I	L	T	G	H	O	M	R	A
L	T	G	H	O	M	R	A	I
T	G	H	O	M	R	A	I	L
G	H	O	M	R	A	I	L	T
H	O	M	R	A	I	L	T	G
O	M	R	A	I	L	T	G	H
M	R	A	I	L	T	G	H	O
R	A	I	L	T	G	H	O	M

S	V	O	G	L	I	N
N	O	I	S	V	G	L
I	N	L	O	G	S	V
L	I	G	V	N	O	S
V	G	S	N	I	L	O
G	S	V	L	O	N	I
O	L	N	I	S	V	S

S	K	O	·	D	U
D	U	S	K	·	O
O	·	D	U	S	K
U	D	K	O	·	S
·	S	U	D	K	O
K	O	·	S	D	U

P	L	E	·	Z	U
Z	U	P	L	E	·
E	·	Z	U	P	L
U	E	L	Z	·	P
·	P	U	E	L	Z
L	Z	·	P	U	E

78. (a) (Solution by A. E. Brouwer, `homepages.cwi.nl/~aeb/games/sudoku/nrc.html`, 2006.) The four new boxes force also `aaaaaaaaa`, ..., `eeeeeeeee` to be rainbows.

e	a	a	a	e	b	b	b	e
c				c				c
c				c				c
c				c				c
e	a	a	a	e	b	b	b	e
d				d				d
d				d				d
d				d				d
e	a	a	a	e	b	b	b	e

(i)

3	2	7	1	5	4	8	6	9
6	1	5	8	2	9	4	7	3
8	9	4	3	7	6	2	1	5
9	6	2	7	1	5	3	8	4
7	4	3	9	8	2	1	5	6
5	8	1	4	6	3	7	9	2
1	3	6	2	9	8	5	4	7
4	7	9	5	3	1	6	2	8
2	5	8	6	4	7	9	3	1

(ii)

3	1	6	8	9	4	7	2	5
5	7	8	3	2	6	4	1	9
4	2	9	5	1	7	3	8	6
7	4	1	6	3	9	2	5	8
2	9	5	7	8	1	6	3	4
8	6	3	4	5	2	9	7	1
9	8	7	1	4	3	5	6	2
6	5	2	9	7	8	1	4	3
1	3	4	2	6	5	8	9	7

(b) Introduce new primary items b'_{yk} for $0 \le y < 9$ and $1 \le k \le 9$. Add b'_{yk} to option (30) with $y = 3\lfloor i\tau/3 \rfloor + \lfloor j\tau/3 \rfloor$, where τ is the permutation (03)(12)(58)(67).

(c) With items b'_{yk} only considered for $y \in \{0, 2, 6, 8\}$, Algorithm D's search tree grows from 76 nodes to 230 for (i), and from 150 nodes to 707 for (ii).

[Puzzle (ii) is a variant of an 11-clue example constructed by Brouwer. The minimum number of clues necessary for hypersudoku is unknown.]

(d) True. (That's the permutation τ in (b), applied to both rows and columns.)

97. (a) $(x \circ y) \circ x = (x \circ y) \circ (y \circ (x \circ y)) = y$.

(b) All five are legitimate. (The last two are gropes because $f(t + f(t)) = t$ for $0 \le t < 4$ in each case; they are isomorphic if we interchange any two elements. The third is isomorphic to the second if we interchange $1 \leftrightarrow 2$. There are 18 grope tables of order 4, of which (4, 12, 2) are isomorphic to the first, third, and last tables shown here.)

(c) For example, let $x \circ y = (-x - y) \bmod n$. (More generally, if G is any group and if $\alpha \in G$ satisfies $\alpha^2 = 1$, we can let $x \circ y = \alpha x^- \alpha y^- \alpha$. If G is commutative and $\alpha \in G$ is arbitrary, we can let $x \circ y = x^- y^- \alpha$.)

(d) For each option of type (i) in an exact covering, define $x \circ x = x$; for each of type (ii), define $x \circ x = y$, $x \circ y = y \circ x = x$; for each of type (iii), define $x \circ y = z$, $y \circ z = x$, $z \circ x = y$. Conversely, every grope table yields an exact covering in this way.

(e) Such a grope covers n^2 items with k options of size 1, all other options of size 3. [F. E. Bennett proved, in *Discrete Mathematics* **24** (1978), 139–146, that such gropes exist for *all* k with $0 \le k \le n$ and $k \equiv n^2$ (modulo 3), except when $k = n = 6$.]

Notes: The identity $x \circ (y \circ x) = y$ seems to have first been considered by E. Schröder in *Math. Annalen* **10** (1876), 289–317 [see '(C_0)' on page 306], but he didn't do much with it. In a class for sophomore mathematics majors at Caltech in 1968, the author defined gropes and asked the students to discover and prove as many theorems about them as they could, by analogy with the theory of groups. The idea was to "grope for results." The official modern term for a grope is a real jawbreaker: *semisymmetric quasigroup*.

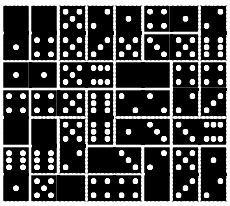
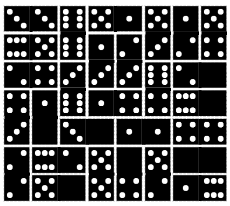
98. (a) Eliminate the n items for xx; use only the $2\binom{n}{3}$ options of type (iii) for which $y \ne z$. (Idempotent gropes are equivalent to "Mendelsohn triples," which are families of $n(n-1)/3$ three-cycles (xyz) that include every ordered pair of distinct elements. N. S. Mendelsohn proved [*Computers in Number Theory* (New York: Academic Press, 1971), 323–338] that such systems exist for all $n \not\equiv 2$ (modulo 3), except when $n = 6$.

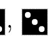
(b) Use only the $\binom{n+1}{2}$ items xy for $0 \le x \le y < n$; replace options of type (ii) by '$xx\ xy$' and '$xy\ yy$' for $0 \le x < y < n$; replace those of type (iii) by '$xy\ xz\ yz$' for $0 \le x < y < z < n$. (Such systems, Schröder's '(C_1) and (C_2)', are called totally symmetric quasigroups; see S. K. Stein, *Trans. Amer. Math. Soc.* **85** (1957), 228–256, §8. If idempotent, they're equivalent to Steiner triple systems.)

(c) Omit items for which $x = 0$ or $y = 0$. Use only the $2\binom{n-1}{3}$ options of type (iii) for $1 \le x < y, z < n$ and $y \ne z$. (Indeed, such systems are equivalent to idempotent gropes on the elements $\{1, \ldots, n-1\}$.)

99. ... (after this dummy answer, notice that the paragraph indentations get bigger)

142. In (a), four pieces change; in (b) the solution is unique:

(a)  ; (b)  .

Notice that the spot patterns , , and  are rotated when a domino is placed vertically; these visual clues, which would disambiguate (a), don't show up in the matrix.

(Dominosa was invented by O. S. Adler [Reichs Patent #71539 (1893); see his booklet *Sperr-Domino und Dominosa* (1912), 23–64, written with F. Jahn]. Similar problems of "quadrilles" had been studied earlier by E. Lucas and H. Delannoy; see Lucas's [*Récréations Mathématiques* **2** (Paris: Gauthier-Villars, 1883), 52–63].)

143. Define 28 vertices Dxy for $0 \le x \le y \le 6$; 28 vertices ij for $0 \le i < 7$, $0 \le j < 8$, and $i + j$ even; and 28 similar vertices ij with $i + j$ odd. The matching problem has 49 triples of the form $\{\mathrm{D}xy, ij, i(j{+}1)\}$ for $0 \le i, j < 7$, as well as 48 of the form $\{\mathrm{D}xy, ij, (i{+}1)j\}$ for $0 \le i < 6$ and $0 \le j < 8$, corresponding to potential horizontal or vertical placements. For example, the triples for exercise 142(a) are $\{\mathrm{D}00, 00, 01\}$, $\{\mathrm{D}05, 01, 02\}$, ..., $\{\mathrm{D}23, 66, 67\}$; $\{\mathrm{D}01, 00, 10\}$, $\{\mathrm{D}04, 01, 11\}$, ..., $\{\mathrm{D}12, 57, 67\}$.

Bennett
Schröder
Caltech
author
groups
quasigroup
semisymmetric quasigroup
Mendelsohn triples
Schröder
totally symmetric quasigroups
Stein
Steiner triple systems
Adler
Jahn
quadrilles
Lucas
Delannoy

144. Model (i) has $M = 56!/8!^7 \approx 4.10 \times 10^{42}$ equally likely possibilities; model (ii) has $N = 1292697 \cdot 28! \cdot 2^{21} \approx 8.27 \times 10^{41}$, because there are 1292697 ways to pack 28 dominoes in a 7×8 frame. (Algorithm D will quickly list them all.) The expected number of solutions per trial in model (i) is therefore $N/M \approx 0.201$.

Ten thousand random trials with model (i) gave 216 cases with at least one solution, including 26 where the solution was unique. The total number $\sum x$ of solutions was 2256; and $\sum x^2 = 95918$ indicated a heavy-tailed distribution whose empirical standard deviation is ≈ 3.1. The total running time was about 250 Mμ.

Ten thousand random trials with model (ii), using random choices from a precomputed list of 1292687 packings, gave 106 cases with a unique solution; one case had 2652 of them! Here $\sum x = 508506$ and $\sum x^2 = 144119964$ indicated an empirical mean of ≈ 51 solutions per trial, with standard deviation ≈ 109. Total time was about 650 Mμ.

158. Before setting $i \leftarrow$ TOP(x_0) in step D6 when $l = 0$, let node x be the spacer at the right of x_0's option, and set $j \leftarrow$ TOP$(x-1)$. If $j > N$ (that is, if that option ends with the secondary item j), cover(j). [This extension applies also to Algorithm C, but one should cover j only if COLOR$(x-1) = 0$.]

160. Let CUTOFF (initially ∞) point to the spacer at the end of the best solution found so far. We'll essentially remove all nodes $>$ CUTOFF from further consideration.

Whenever a solution is found, let node PP be the spacer at the end of the option for which $x_k = \max(x_0, \ldots, x_{l-1})$. If PP $\neq$ CUTOFF, set CUTOFF $\leftarrow$ PP, and for $0 \le k < l$ remove all node $>$ CUTOFF from the list for TOP(x_k). (It's easy to do this because the list is sorted.) Minimax solutions follow the last change to CUTOFF.

Begin the subroutine 'uncover(i)' by removing all node $>$ CUTOFF from item i's list. After setting $d \leftarrow$ DLINK(q) in unhide(p), set DLINK$(q) \leftarrow d \leftarrow x$ if $d >$ CUTOFF. Make the same modifications also to the subroutine 'unpurify(p)'.

Subtle point: Suppose we're uncovering item i and encounter an option '$i\ j\ \ldots$' that should be restored to the list of item j; and suppose that the original successor '$j\ a\ \ldots$' of that option for item j lies below the cutoff. We know that '$j\ a\ \ldots$' contains at least one primary item, and that every primary item was covered before we changed the cutoff. Hence '$j\ a\ \ldots$' was *not* restored, and we needn't worry about removing it. We merely need to correct the DLINK, as stated above.

161. Now let CUTOFF be the spacer just *before* the best solution known. When resetting CUTOFF, backtrack to level $k-1$, where x_k maximizes $\{x_0, \ldots, x_{l-1}\}$.

164. Use $2n$ primary items a_i, d_j for the "across" and "down" words, together with n^2 secondary items ij for the individual cells. Also use W secondary items w, one for each legal word. The cover problem has $2Wn$ options, namely '$a_i\ i1{:}c_1\ \ldots\ in{:}c_n\ c_1\ldots c_n$' and '$d_j\ 1j{:}c_1\ \ldots\ nj{:}c_n\ c_1\ldots c_n$' for $1 \le i, j \le n$ and each legal word $c_1 \ldots c_n$.

We can avoid having both a solution and its transpose by introducing W further secondary items w' and appending $c_1 \ldots c_n'$ at the right of each option for a_1 and d_1. Then exercise 158's variant of Algorithm C will never choose a word for d_1 that it has already tried for a_1. (Think about it.)

But this construction is *not* a win for "dancing links," because it causes massive amounts of data to go in and out of the active structure. For example, with the five-letter words of WORDS(5757), it correctly finds all 323,264 of the double word squares, but its running time is 15 *tera*mems! Much faster is to use the algorithm of exercise 7.2.2–28, which needs only 46 gigamems to discover all of the 1,787,056 unrestricted word squares; the double word squares are easily identified among those solutions.

dimer tilings
heavy-tailed distribution
empirical standard deviation
color constraints
CUTOFF
sorted

binary search
minimax
minimax
minimax
Flower Power
Shortz
Incroci Concentrici
Petal Pushers

165. One could do a binary search, trying varying values of W. But the best way is to use the construction of exercise 164 together with the minimax variant of Algorithm C in exercise 160. This works perfectly, when the options for most common words come first.

Indeed, this method finds the double square '`BLAST|EARTH|ANGER|SCOPE|TENSE`' and proves it best in just 64 Gμ, almost as fast as the specialized method of exercise 7.2.2–28. (That square contains `ARGON`, the 1720th most common five-letter word, in its third column; the next-best squares use `PEERS`, which has rank 1800.)

166. The "minimax" method of exercise 165 finds the first five squares of

```
                                              H E R T Z E S
                              C H E S T S     O P E R A T E
                  S T A R T   L U S T R E     M I M I C A L
        S H O W   T H R E E   O B T A I N     A C E R A T E
  M A Y N O N E   R O O F S   A R E N A S     G E N E T I C
IS A G E O P E N   A S S E T   C I R C L E     E N D M O S T
TO N O T W E S T   P E E R S   A S S E S S     R E S E N T S
```

IS TO	MAY AGE NOT	SHOW NONE OPEN WEST	START THREE ROOFS ASSET PEERS	CHESTS LUSTRE OBTAIN ARENAS CIRCLE ASSESS	HERTZES OPERATE MIMICAL ACERATE GENETIC ENDMOST RESENTS

in respectively 200 Kμ, 15 Mμ, 450 Mμ, 25 Gμ, 25.6 Tμ. It struggles to find the best 6×6, because too few words are cut off from the search; and it thrashes miserably with the 24 thousand 7-letter words, because those words yield only seven extremely esoteric solutions. For those lengths it's best to cull the 2038753 and 14513 *unrestricted* word squares, which the method of exercise 7.2.2–28 finds in respectively 4.6 Tμ and 8.7 Tμ.

168. An exact cover problem with colors, as in answer 64, works nicely: There are $2p$ primary items a_i and d_i for the final words, and $pn + W$ secondary items ij and w for the cells and potential words, where $0 \le i < p$ and $1 \le j \le n$. The Wp options going across are 'a_i $i1{:}c_1$ $i2{:}c_2$ $\ldots$ $in{:}c_n$ $c_1 \ldots c_n$'. The Wp options going down are 'd_i $i1{:}c_1$ $((i{+}1) \bmod p)2{:}c_2$ $\ldots$ $((i{+}n{-}1) \bmod p)n{:}c_n$ $c_1 \ldots c_n$' for left-leaning stairs; 'd_i $i1{:}c_n$ $((i{+}1) \bmod p)2{:}c_{n-1}$ $\ldots$ $((i{+}n{-}1) \bmod p)n{:}c_1$ $c_1 \ldots c_n$' for right-leaning stairs. The modification to Algorithm C in exercise 158 saves a factor of $2p$; and the minimax modification in exercise 160 hones in quickly on optimum solutions.

There are no left word stairs for $p = 1$, since we need two distinct words. The left winners for $2 \le p \le 10$ are: '`WRITE|WHOLE`'; '`MAKES|LIVED|WAXES`'; '`THERE|SHARE|WHOLE|WHOSE`'; '`STOOD|THANK|SHARE|SHIPS|STORE`'; '`WHERE|SHEEP|SMALL|STILL|WHOLE|SHARE`'; '`MAKES|BASED|TIRED|WORKS|LANDS|LIVES|GIVES`'; '`WATER|MAKES|LOVED|GIVES|LAKES|BASED|NOTES|BONES`'; '`WHERE|SHEET|STILL|SHALL|WHITE|SHAPE|STARS|WHILE|SHORE`'; '`THERE|SHOES|SHIRT|STONE|SHOOK|START|WHILE|SHELL|STEEL|SHARP`'. They all belong to `WORDS(500)`, except that $p = 8$ needs `WORDS(504)` for `NOTED`.

The right winners have a bit more variety: '`SPOTS`'; '`STALL|SPIES`'; '`STOOD|HOLES|LEAPS`'; '`MIXED|TEARS|SLEPT|SALAD`'; '`YEARS|STEAM|SALES|MARKS|DRIED`'; '`STEPS|SEALS|DRAWS|KNOTS|TRAPS|DROPS`'; '`TRIED|FEARS|SLIPS|SEAMS|DRAWS|ERECT|TEARS`'; '`YEARS|STOPS|HOOKS|FRIED|TEARS|SLANT|SWORD|SWEEP`'; '`START|SPEAR|SALES|TESTS|STEER|SPEAK|SKIES|SLEPT|SPORT`'; '`YEARS|STOCK|HORNS|FUELS|BEETS|SPEED|TEARS|PLANT|SWORD|SWEEP`'. They belong to `WORDS(1300)` except when p is 2 or 3.

[Arrangements equivalent to left word stairs were introduced in America under the name "Flower Power" by Will Shortz in *Classic Crossword Puzzles* (Penny Press, February 1976), based on Italian puzzles called "Incroci Concentrici" in *La Settimana Enigmistica*. Shortly thereafter, in *GAMES* magazine and with $p = 16$, he called them "Petal Pushers," usually based on six-letter words but occasionally going to seven. Left word stairs are much more common than the right-leaning variety, because the latter mix end-of-word with beginning-of-word letter statistics.]

kernels
induced subgraph
in-degree
out-degree
strong components

169. Consider all "kernels" $c_1 \ldots c_{14}$ that can appear as illustrated, within a right word stair of 5-letter words. Such kernels arise for a given set of words only if there are letters $x_1 \ldots x_{12}$ such that $x_3x_4x_5c_2c_3$, $c_4c_5c_6c_7c_8$, $c_9c_{10}c_{11}c_{12}x_6$, $c_{13}c_{14}x_7x_8x_9$, $x_1x_2x_5c_5c_9$, $c_1c_2c_6c_{10}c_{13}$, $c_3c_7c_{11}c_{14}x_{10}$, and $c_8c_{12}x_7x_{11}x_{12}$ are all in the set. Thus it's an easy matter to set up an exact cover problem (with colors) that will find the multiset of kernels, after which we can extract the set of *distinct* kernels.

$$\begin{array}{cccccccc}
 & & x_1 & & & & & \\
 & & x_2 & c_1 & & & & \\
x_3 & x_4 & x_5 & c_2 & c_3 & & & \\
 & c_4 & c_5 & c_6 & c_7 & c_8 & & \\
 & & c_9 & c_{10} & c_{11} & c_{12} & x_6 & \\
 & & & c_{13} & c_{14} & x_7 & x_8 & x_9 \\
 & & & & x_{10} & x_{11} & & \\
 & & & & & x_{12} & &
\end{array}$$

Construct the digraph whose arcs are the kernels, and whose vertices are the 9-tuples that arise when kernel $c_1 \ldots c_{14}$ is regarded as the transition

$$c_1c_2c_3c_4c_5c_6c_7c_9c_{10} \;\to\; c_3c_7c_8c_9c_{10}c_{11}c_{12}c_{13}c_{14}.$$

This transition contributes two words, $c_4c_5c_6c_7c_8$ and $c_1c_2c_6c_{10}c_{13}$, to the word stair. Indeed, *right word stairs of period p are precisely the p-cycles in this digraph for which the $2p$ contributed words are distinct.*

Now we can solve the problem, if the graph isn't too big. For example, `WORDS`(1000) leads to a digraph with 180524 arcs and 96677 vertices. We're interested only in the oriented cycles of this (very sparse) digraph; so we can reduce it drastically by looking only at the largest induced subgraph for which each vertex has positive in-degree and positive out-degree. (See exercise 7.1.4–??, where a similar reduction was made.) And wow: That subgraph has only 30 vertices and 34 arcs! So it is totally understandable, and we deduce quickly that the longest right word stair belonging to `WORDS`(1000) has $p = 5$. That word stair, which we found directly in answer 68, corresponds to the cycle

`SEDYEARST` $\to$ `DRSSTEASA` $\to$ `SAMSALEMA` $\to$ `MESMARKDR` $\to$ `SKSDRIEYE` $\to$ `SEDYEARST`.

A similar approach applies to left word stairs, but the kernel configurations are reflected left-to-right; transitions then contribute the words $c_8c_7c_6c_5c_4$ and $c_1c_2c_6c_{10}c_{13}$. The digraph from `WORDS`(500) turns out to have 136771 arcs and 74568 vertices; but this time 6280 vertices and 13677 arcs remain after reduction. Decomposition into strong components makes the task simpler, because every cycle belongs to a strong component. Still, we're stuck with a giant component that has 6150 vertices and 12050 arcs.

The solution is to reduce the current subgraph repeatedly as follows: Find a vertex v of out-degree 1. Backtrack to discover a simple path, from v, that contributes only distinct words. If there is no such path (and there usually isn't, and the search usually terminates quickly), remove v from the graph and reduce it again.

With this method one can rapidly show that the longest left word stair from `WORDS`(500) has period length 36: '`SHARE|SPENT|SPEED|WHEAT|THANK|CHILD|SHELL|SHORE|STORE|STOOD|CHART|GLORY|FLOWS|CLASS|NOISE|GAMES|TIMES|MOVES|BONES|WAVES|GASES|FIXED|TIRED|FEELS|FALLS|WORLD|ROOMS|WORDS|DOORS|PARTY|WANTS|WHICH|WHERE|SHOES|STILL|STATE`', with 36 other words that go down. Incidentally, `GLORY` and `FLOWS` have ranks 496 and 498, so they just barely made it into `WORDS`(500).

Larger values of W are likely to lead to quite long cycles from `WORDS`(W). Their discovery won't be easy, but the search will no doubt be instructive.

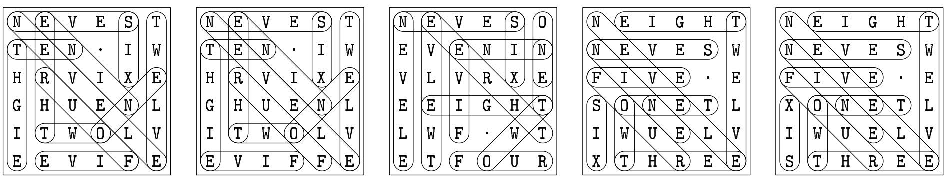
170. Use $3p$ primary items a_i, b_i, d_i for the final words; $pn + 2W$ secondary items ij, w, w' for the cells and potential words, with $0 \le i < p$ and $1 \le j \le n$ (somewhat as in answer 68). The Wp options going across are 'a_i $i1{:}c_1$ $i2{:}c_2$ $\ldots$ $in{:}c_n$ $c_1 \ldots c_n$ $c_1 \ldots c_n'$'. The $2Wp$ options going down in each way are 'b_i $i1{:}c_1$ $((i{+}1) \bmod p)2{:}c_2$ $\ldots$ $((i{+}n{-}1) \bmod p)n{:}c_n$ $c_1 \ldots c_n$' and 'd_i $i1{:}c_n$ $((i{+}1) \bmod p)2{:}c_{n-1}$ $\ldots$ $((i{+}n{-}1) \bmod p)n{:}c_1$ $c_1 \ldots c_n$'. The items w' at the right of the a_i options save us a factor of p.

Use Algorithm C (modified). We can't have $p = 1$. Then comes '`SPEND|SPIES`'; '`WAVES|LINED|LEPER`'; '`LOOPS|POUTS|TROTS|TOONS`'; '`SPOOL|STROP|STAID|SNORT|SNOOT`'; '`DIMES|MULES|RIPER|SIRED|AIDED|FINED`'; '`MILES|LINTS|CARES|LAMED|PIPED|SANER|LIVER`'; '`SUPER|ROVED|TILED|LICIT|CODED|ROPED|TIMED|DOMED`'; '`FORTH|LURES|MIRES|POLLS|SLATS|SPOTS|SOAPS|PLOTS|LOOTS`'; '`TIMES|FUROR|RUNES|MIMED|CAPED|PACED|LAVER|FINES|LIMED|MIRES`'. (Lengthy computations were needed for $p \geq 8$.)

171. Now $p \leq 2$ is impossible. A construction like the previous one allows us again to save a factor of p. (There's also top/bottom symmetry, but it is somewhat harder to exploit.) Examples are relatively easy to find, and the winners are '`MILES|GALLS|BULLS`'; '`FIRES|PONDS|WALKS|LOCKS`'; '`LIVES|FIRED|DIKES|WAVED|TIRES`'; '`BIRDS|MARKS|POLES|WAVES|WINES|FONTS`'; '`LIKED|WARES|MINES|WINDS|MALES|LOVES|FIVES`'; '`WAXES|SITES|MINED|BOXES|CAVES|TALES|WIRED|MALES`'; '`CENTS|HOLDS|BOILS|BALLS|MALES|WINES|FINDS|LORDS|CARES`'; '`LOOKS|ROADS|BEATS|BEADS|HOLDS|COOLS|FOLKS|WINES|GASES|BOLTS`'. [Such patterns were introduced by Harry Mathews in 1975, who gave the four-letter example '`TINE|SALE|MALE|VINE`'. See H. Mathews and A. Brotchie, *Oulipo Compendium* (London: Atlas, 1998), 180–181.]

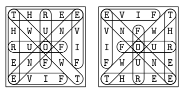
175. Given a 3SAT problem with clauses $(l_{i1} \vee l_{i2} \vee l_{i3})$ for $1 \leq i \leq m$, with each $l_{ij} \in \{x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n\}$, construct an exact cover problem with $3m$ primary items ij $(1 \leq i \leq m,\ 1 \leq j \leq 3)$ and n secondary items x_k $(1 \leq k \leq n)$, having the following options: (i) '$l_{i1}\ l_{i2}$', '$l_{i2}\ l_{i3}$', '$l_{i3}\ l_{i1}$'; (ii) '$l_{ij}\ x_k$:1' if $l_{ij} = x_k$, '$l_{ij}\ x_k$:0' if $l_{ij} = \bar{x}_k$. That problem has a solution if and only if the given clauses are satisfiable.

180. There are just five solutions; the latter two are flawed by being disconnected:



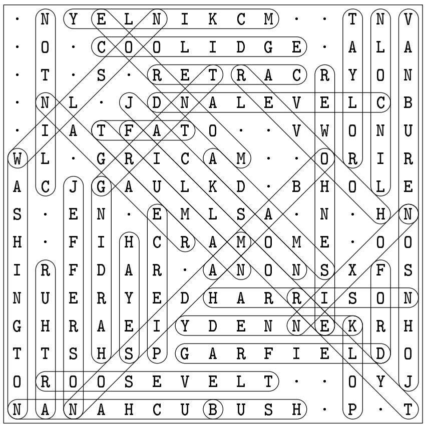
Historical note: Word search puzzles were invented by Norman E. Gibat in 1968.

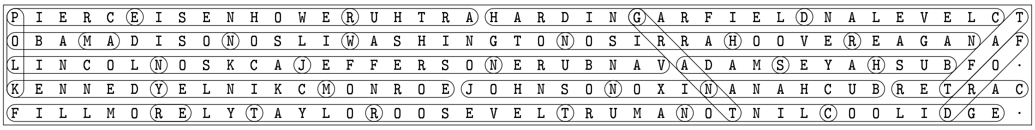
181. When Algorithm C is generalized to allow non-unit item sums as in Algorithm M, it needs just 24 megamems to prove that there are exactly eight solutions — which all are rotations of the two shown here.



Mathews
Brotchie
Oulipo
3SAT
disconnected
Gibat

182. (a, b) The author's best solutions, thought to be minimal (but there is no proof), are below. In both cases, and in Fig. 71, an interactive method was used: After the longest words were placed strategically by hand, Algorithm C packed the others nicely.

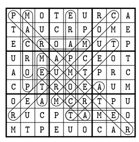




[Solution (b) applies an idea by which Leonard Gordon was able to pack the names of presidents 1–42 with one less column. See A. Ross Eckler, *Word Ways* **27** (1994), 147; see also page 252, where `OBAMA` miraculously fits into Gordon's 15×15 solution!]

183. To pack w given words, use primary items $\{\mathrm{P}ij, \mathrm{R}ic, \mathrm{C}ic, \mathrm{B}ic, \#k \mid 1 \le i, j \le 9,\ 1 \le k \le w,\ c \in \{\mathtt{A}, \mathtt{C}, \mathtt{E}, \mathtt{M}, \mathtt{O}, \mathtt{P}, \mathtt{R}, \mathtt{T}, \mathtt{U}\}\}$ and secondary items $\{ij \mid 1 \le i, j \le 9\}$. There are 729 options '$\mathrm{P}ij$ $\mathrm{R}ic$ $\mathrm{C}jc$ $\mathrm{B}bc$ $ij{:}c$', where $b = 3\lfloor (i-1)/3\rfloor + \lceil j/3\rceil$, together with an option '$\#k$ $i_1j_1{:}c_1$ $\ldots i_lj_l{:}c_l$' for each placement of an l-letter word $c_1 \ldots c_l$ into cells $(i_1, j_1), \ldots, (i_l, j_l)$. Furthermore, it's important to *modify* step C3 of the algorithm so that the "best item" i always has the form $\#k$, unless $\mathtt{LEN}(i) \le 1$.

A brief run then establishes that `COMPUTER` and `CORPORATE` cannot both be packed. But all of the words *except* `CORPORATE` do fit together; the (unique) solution shown is found after only 7.3 megamems, most of which are needed simply to input the problem. [This exercise was inspired by a puzzle in *Sudoku Masterpieces* (2010) by Huang and Snyder.]



author
interactive method
Gordon
Eckler
branch, choice of
choice of item to cover
sharp preference
best item
Huang
Snyder

sharp preference heuristic

185. To pack w given words, use $w + m(n-1) + (m-1)n$ primary items $\{\#k \mid 1 \le k \le w\}$ and $\{\mathrm{H}ij, \mathrm{V}ij \mid 1 \le i \le m,\ 1 \le j \le n\}$, but with H$in$ and Vmj omitted; Hij represents the edge between cells (i,j) and $(i,j+1)$, and Vij is similar. There also are $2mn$ secondary items $\{ij, ij' \mid 1 \le i \le m,\ 1 \le j \le n\}$. Each horizontal placement of the kth word $c_1 \ldots c_l$ into cells $(i,j+1), \ldots, (i,j+l)$ generates the option

$$\#k\ ij{:}.\ \ ij'{:}\mathtt{0}\ \ i(j{+}1){:}c_1\ \ i(j{+}1)'{:}\mathtt{1}\ \ \mathrm{H}i(j{+}1)\ \ i(j{+}2){:}c_2\ \ i(j{+}2)'{:}\mathtt{1}\ \ \mathrm{H}i(j{+}2)\ \ldots$$
$$\mathrm{H}i(j{+}l{-}1)\ \ i(j{+}l){:}c_l\ \ i(j{+}l)'{:}\mathtt{1}\ \ i(j{+}l{+}1){:}.\ \ i(j{+}l{+}1)'{:}\mathtt{0}$$

with $3l+4$ items, except that '$ij{:}.\ ij'{:}\mathtt{0}$' is omitted when $j = 0$ and '$i(j{+}l{+}1){:}.\ i(j{+}l{+}1)'{:}\mathtt{0}$' is omitted when $j{+}l = n$. Each vertical placement is similar. For example,

$$\#1\ \ 11{:}\mathtt{Z}\ \ 11'{:}\mathtt{1}\ \ \mathrm{V}11\ \ 21{:}\mathtt{E}\ \ 21'{:}\mathtt{1}\ \ \mathrm{V}21\ \ 31{:}\mathtt{R}\ \ 31'{:}\mathtt{1}\ \ \mathrm{V}31\ \ 41{:}\mathtt{0}\ \ 41'{:}\mathtt{1}\ \ 51{:}.\ \ 51'{:}\mathtt{0} \qquad (*)$$

is the first vertical placement option for `ZERO`, if `ZERO` is word #1. When $m = n$, however, we save a factor of 2 by omitting all of the vertical placements of word #1.

To enforce the tricky condition (ii), we also introduce $3m(n-1) + 3(m-1)n$ options:

$$\begin{array}{ll}
\mathrm{H}ij\ \ ij'{:}\mathtt{0}\ \ i(j{+}1)'{:}\mathtt{1}\ \ ij{:}. & \mathrm{V}ij\ \ ij'{:}\mathtt{0}\ \ (i{+}1)j'{:}\mathtt{1}\ \ ij{:}. \\
\mathrm{H}ij\ \ ij'{:}\mathtt{1}\ \ i(j{+}1)'{:}\mathtt{0}\ \ i(j{+}1){:}. & \mathrm{V}ij\ \ ij'{:}\mathtt{1}\ \ (i{+}1)j'{:}\mathtt{0}\ \ (i{+}1)j{:}. \\
\mathrm{H}ij\ \ ij'{:}\mathtt{0}\ \ i(j{+}1)'{:}\mathtt{0}\ \ ij{:}.\ \ i(j{+}1){:}. & \mathrm{V}ij\ \ ij'{:}\mathtt{0}\ \ (i{+}1)j'{:}\mathtt{0}\ \ ij{:}.\ \ (i{+}1)j{:}.
\end{array}$$

This construction works nicely because each edge must encounter either a word that crosses it or a space that touches it. (Beware of a slight glitch: A valid solution to the puzzle might have several compatible choices for Hij and Vij in "blank" regions.) *Important:* The change to step C3 in answer 83, which branches only on $\#k$ items unless an H or V is forced, should be followed here because it gives an enormous speedup.

The cover problem for our 11-word example has 1192 options, 123 + 128 items, and 9127 solutions, found in 29 Gμ. But only 20 of those solutions are connected; and they yield only the three distinct word placements below. A slightly smaller rectangle, 7×9, also has three valid placements. The smallest rectangle that admits a solution to (i) and (ii) is 5×11; that placement is *unique*, but it has two components:

```
      F     F  
Z E R O   S I X
  I   U     V  
  G   R   T E N
  H       H   I
  T W O   R   N
      N   E   E
    S E V E N  
```

```
    F   T W O  
    O N E     S
    U   N I N E
Z E R O       V
  I     F     E
  G   S I X   N
  H     V      
  T H R E E    
```

```
    F I V E    
T W O     I    
    U     G    
Z E R O   H    
      N   T E N
S E V E N     I
I             N
X     T H R E E
```

```
  E   T            
F I V E     S I X
  G   N I N E    
  H         V   F
  T W O   Z E R O
      N     N   U
T H R E E       R
```

```
F   S I X   F   T    
O N E     E I G H T  
U   V   T   V   R   T
R   E   E   E   E   W
    N I N E   Z E R O
```

Instead of generating all solutions to (i) and (ii) and discarding the disconnected ones, there's a much faster way to guarantee connectedness throughout the search; but it requires major modifications to Algorithm C. Whenever no H or V is forced, we can list all active options that are connected to word #1 and not smaller than choices that could have been made earlier. Then we branch on them, instead of branching on an item. For example, if (*) above is used to place `ZERO`, it will force H00 and H20 and V30. The next decision will be to place either `EIGHT` or `ONE`, in the places where they overlap `ZERO`. (However, we'll be better off if we order the words by decreasing length, so that for instance #1 is `EIGHT` and #11 is `ONE`.) Interested readers are encouraged to work out the instructive details. This method needs only 630 Mμ to solve the example problem, as it homes right in on the three connected solutions.

McDonald
Gordon
Russell

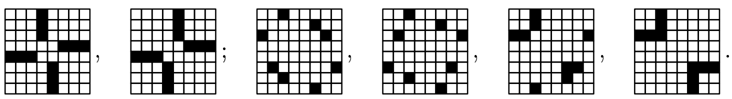
186. Gary McDonald found this remarkable 20×20 solution in 2017:

```
W I L S O N   T A F T               P
          I   A   O   C   J O H N S O N
          X   Y   R   L   E         L
T   H   C O O L I D G E   F   A     K
R E A G A N   O       V   F   R
U   R   R     R O O S E V E L T
M   R   T   B     B   L   R   H A Y E S
A   I   E   U     A D A M S   U     I
N   S   R   C     M   N   O   R     S
    O       H   H A R D I N G       E
    N       A   O             G R A N T
F       M O N R O E   J             H
I           A   V   W A S H I N G T O N
L   M C K I N L E Y   C         A   W
L     L         R     K   P I E R C E
M A D I S O N     B U S H       F   R
O     N               O         I
R     T Y L E R   V A N   B U R E N
E     O                         L
  L I N C O L N       K E N N E D Y
```

A 19×19 is surely impossible, although no proof is known. L. Gordon had previously fit the names of presidents 1–42 into an 18×22 rectangle [*Word Ways* **27** (1994), 63].

187. (a) Set up an exact cover problem as in answer 185, but with just three words `AAA`, `AAAA`, `AAAAA`; then adjust the multiplicities and apply Algorithm M. The two essentially distinct answers are shown below; one of them is disconnected, hence disqualified.

(b) Similarly, we find four essentially different answers, only two of which are OK:



Algorithm M handles case (b) with ease (5 Gμ). But it does not explore the space of possibilities for case (a) intelligently, and costs 591 Gμ.

190. Besides the primary items p_{ij}, r_{ik}, c_{jk}, b_{xk} of (30), introduce R_{ik}, C_{jk}, and B_{xk} for the permuted array, as well as u_k and v_l to define a permutation. Also introduce secondary items π_k to record the permutation and ij to record the value at cell (i,j). The permutation is defined by 81 options 'u_k v_l π_k:l' for $1 \le k, l \le 9$. And there are $9^4 = 6561$ other options, one for each cell (i,j) of the board and each pair (k,l) of values before and after α is applied. If $(ij)\alpha = i'j'$, let $x' = 3\lfloor i'/3 \rfloor + \lfloor j'/3 \rfloor$. Then option (i,j,k,l) is normally 'p_{ij} r_{ik} c_{jk}, b_{xk}, $R_{i'l}$, $C_{j'l}$, $B_{x'l}$, ij:k $i'j'$:l, π_k:l'. However, if $i' = i$ and $j' = j$, that option is shortened to 'p_{ij} r_{ik} c_{jk}, b_{xk}, $R_{i'l}$, $C_{j'l}$, $B_{x'l}$, ij:k π_k:l', and it is omitted when $k \ne l$. The options $(0,j,k,l)$ are also omitted when $k \ne j+1$, in order to force '`123456789`' on the top row.

With that top row and with α = transposition, Algorithm C produces 30,258,432 solutions in 2.2 teramems. (These solutions were first enumerated in 2005 by E. Russell; see `www.afjarvis.staff.shef.ac.uk/sudoku/sudgroup.html`.)

191. A similar method applies, but with additional items b'_{yk} and $B'_{y'l}$ as in exercise 78. The number of solutions is (a) 7784; (b) 16384; (c) 372; (d) 32. Here are examples of (a) and (d); the latter is shown with labels $\{0, \ldots, 7, *\}$, to clarify its structure.

[Enumerations (a), (b), (c) were first carried out by Bastian Michel in 2007.]

(a)

1	2	3	4	5	6	7	8	9
9	7	4	3	1	8	5	6	2
8	5	6	9	7	2	1	3	4
5	8	2	1	3	9	4	7	6
4	1	7	8	6	5	2	9	3
6	3	9	2	4	7	8	5	1
7	4	1	5	9	3	6	2	8
3	6	8	7	2	4	9	1	5
2	9	5	6	8	1	3	4	7

;

(d)

7	0	2	5	1	3	*	4	6
5	3	6	*	7	4	2	0	1
*	1	4	2	0	6	7	5	3
2	5	7	0	6	1	3	*	4
0	4	3	7	*	5	1	6	2
6	*	1	3	4	2	5	7	0
1	7	5	4	2	0	6	3	*
3	2	0	6	5	*	4	1	7
4	6	*	1	3	7	0	2	5

.

200. (a) To cover 2 of 4, we have 3 choices at the root, then 3 or 2 or 1 at the next level, hence (1, 3, 6) cases at levels (0, 1, 2). To cover 5 of 7, there are (1, 3, 6, 10, 15, 21) cases at levels (0, 1, ..., 5). Thus the search profile with item 1 first is (1, 3, 6, $6\cdot3$, $6\cdot6$, $6\cdot10$, $6\cdot15$, $6\cdot21$). The other way is better: (1, 3, 6, 10, 15, 21, $21\cdot3$, $21\cdot6$).

(b) With item 1 first the profile is $(a_0,\ a_1,\ \ldots,\ a_p,\ a_pa_1,\ \ldots,\ a_pa_q)$, where $a_j = \binom{j+d}{d}$. We should branch on item 2 first because $a_{p+1} < a_pa_1$, $a_{p+2} < a_pa_2$, ..., $a_q < a_pa_{q-p}$, $a_qa_1 < a_pa_{q-p+1}$, ..., $a_qa_{p-1} < a_pa_{q-1}$. (These inequalities follow because the sequence $\langle a_j\rangle$ is strongly log-concave: It satisfies the condition $a_j^2 > a_{j-1}a_{j+1}$ for all $j \ge 1$. See exercise MPR–125.)

240. Let the given shape be specified as a set of integer pairs (x, y). These pairs might simply be listed one by one in the input; but it's much more convenient to accept a more compact specification. For example, the utility program with which the author prepared the examples of this book was designed to accept UNIX-like specifications such as '`[14-7]2 5[0-3]`' for the eight pairs $\{(1,2),\ (4,2),\ (5,2),\ (6,2),\ (7,2),\ (5,0),\ (5,1),\ (5,3)\}$. (Notice that a pair is included only once, if it's specified more than once.) The range $0 \le x, y < 62$ has proved to be sufficient in almost all instances, with such integers encoded as single "extended hexadecimal digits" `0`, `1`, ..., `9`, `a`, `b`, ..., `z`, `A`, `B`, ..., `Z`. The specification '`[1-3][1-k]`' is one way to define a 3×20 rectangle.

Similarly, each of the given polyominoes is specified by stating its piece name and a set T of typical positions that it might occupy. Such positions (x, y) are specified using the same conventions that were used for the shape; they needn't lie within that shape.

The program computes *base placements* by rotating and/or reflecting the elements of that set T. The first base placement is the shifted set $T_0 = T - (x_{\min}, y_{\min})$, whose coordinates are nonnegative and as small as possible. Then it repeatedly applies an elementary transformation, either $(x, y) \mapsto (y, x_{\max} - x)$ or $(x, y) \mapsto (y, x)$, to every existing base placement, until no further placements arise. (That process becomes easy when each base placement is represented as a sorted list of packed integers $(x \ll 16) + y$.) For example, the typical positions of the straight tromino might be specified as '`1[1-3]`'; it will have two base placements, $\{(0,0),(0,1),(0,2)\}$ and $\{(0,0),(1,0),(2,0)\}$.

After digesting the input specifications, the program defines the items of the exact problem, which are (i) the piece names; (ii) the cells xy of the given shape.

Finally, it defines the options: For each piece p and for each base placement T' of p, and for each offset (δ_x, δ_y) such that $T' + (\delta_x, \delta_y)$ lies fully within the given shape, there's an option that names the items $\{p\} \cup \{(x+\delta_x, y+\delta_y) \mid (x,y) \in T'\}$.

(The output of this program is often edited by hand, to take account of special circumstances. For example, some items may change from primary to secondary; some options may be eliminated in order to break symmetry. The author's implementation also allows the specification of secondary items with color controls, along with base placements that include such controls.)

Michel
profile
log-concave
author
UNIX
extended hexadecimal digits
hexadecimal notation, extended
base placements
sorted
packed integers
straight tromino
secondary
break symmetry
author
color controls

248. RUSTY. [Leigh Mercer posed a similar question to Martin Gardner in 1960.]

250. As in the 3×20 example considered in the text, we can set up an exact cover problem with $12 + 60$ items, and with options for every potential placement of each piece. This gives respectively (52, 292, 232, 240, 232, 120, 146, 120, 120, 30, 232, 120) options for pieces (O, P, ..., Z) in Conway's nomenclature, thus 1936 options in all.

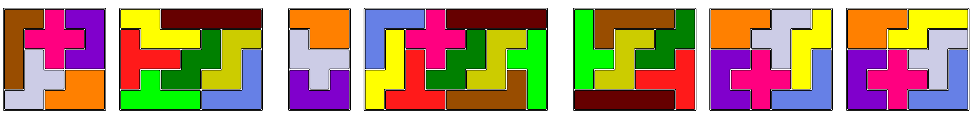
To reduce symmetry, we can insist that the X occurs in the upper left corner; then it contributes just 10 options instead of 30. But some solutions are still counted twice, when X is centered in the middle row. To prevent this we can add a *secondary item* 's': Append 's' to the five options that correspond to those centered appearances; also append 's' to the 60 options that correspond to placements where the Z is flipped over.

Without those changes, Algorithm D would use 9.76 Gμ to find 4040 solutions; with them, it needs just 2.86 Gμ to find 1010.

This approach to symmetry breaking in pentomino problems is due to Dana Scott [Technical Report No. 1 (Princeton University Dept. of Electrical Engineering, 10 June 1958)]. Another way to break symmetry would be to allow X anywhere, but to restrict the W to its 30 *unrotated* placements. That works almost as well: 2.87 Gμ.

251. There's a unique way to pack P, Q, R, U, X into a 5×5 square, and to pack the other seven into a 5×7. (See below.) With independent reflections, together with rotation of the square, we obtain 16 of the 1010. There's also a unique way to pack P, R, U into a 5×3 and the others into a 5×9 (noticed by R. A. Fairbairn in 1967), yielding 8 more. And there's a unique way to pack O, Q, T, W, Y, Z into a 5×6, plus two ways to pack the others, yielding another 16. (These paired 5×6 patterns were apparently first noticed by J. Pestiau; see answer 269.) Finally, the packings in the next exercise give us 264 decomposable 5×12s altogether.

[Similarly, C. J. Bouwkamp discovered that S, V, T, Y pack uniquely into a 4×5, while the other eight can be put into an 4×10 in five ways, thus accounting for 40 of the 368 distinct 4×15s. See *JRM* **3** (1970), 125.]



252. Without symmetry reduction, 448 solutions are found in 1.21 Gμ. But we can restrict X to the upper left corner, as in answer 250, flagging its placements with 's' when centered in the middle row or middle column (but not both). Again the 's' is appended to flipped Z's. Finally, when X is placed in dead center, we append *another* secondary item 'c', and append 'c' to the 90 rotated placements of W. This yields 112 solutions, after 0.34 Gμ.

Or we could leave X unhindered but curtail W to 1/4 of its placements. That's easier to do (although not *quite* as clever) and it finds those 112 in 0.42 Gμ.

Incidentally, there *aren't* actually any solutions with X in dead center.

254. The exact cover problem analogous to that in exercise 250 has $12 + 60$ items and (56, 304, 248, 256, 248, 128, 1152, 128, 128, 32, 248, 128) options. It finds 9356 solutions after 15.93 Gμ of computation, without symmetry reduction. But if we insist that X be centered in the upper left quarter, by removing all but 8 of its placements, we get 2339 solutions after just 3.93 Gμ. (The alternative of restricting W's rotations is *not* as effective in this case: 5.43 Gμ.) These solutions were first enumerated by C. B. and Jenifer Haselgrove [Eureka: *The Archimedeans' Journal* **23** (1960), 16–18].

Mercer
Gardner
Conway
secondary item
flipped over
Scott
break symmetry
Fairbairn
Pestiau
Bouwkamp
flipped Z's
Haselgrove, Colin
Haselgrove, Jenifer

255. (a) Obviously only $k = 5$ is feasible. All such packings can be obtained by omitting all options of the cover problem that straddle the "cut." That leaves 1507 of the original 2032 options, and yields 16 solutions after 104 Mμ. (Those 16 boil down to just the two 5×6 decompositions that we already saw in answer 251.)

(b) Now we remove the 763 options for placements that don't touch the boundary, and obtain just the two solutions below, after 100 Mμ. (This result was first noticed by Tony Potts, who posted it to Martin Gardner on 9 February 1960.)

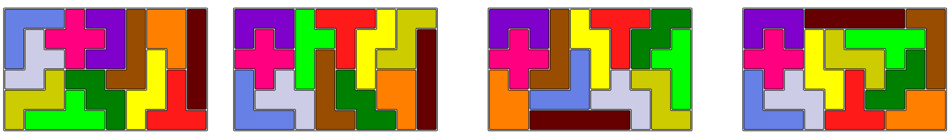
(c) With 1237 placements/options, the *unique* solution is now found after 83 Mμ.

(d) There are respectively (0, 9, 3, 47, 16, 8, 3, 1, 30, 22, 5, 11) solutions for pentominoes (O, P, Q, ..., Z). (The I/O pentomino can be "framed" by the others in 11 ways; but all of those packings also have at least one other interior pentomino.)

(e) Despite many ways to cover all boundary cells with just seven pentominoes, none of them lead to an overall solution. Thus the minimum is eight; 207 of the 2339 solutions attain it. To find them we might as well generate and examine all 2339.

(f) The question is ambiguous: If we're willing to allow the X to touch unnamed pieces at a corner, but not at an edge, there are 25 solutions (8 of which happen to be answers to part (a)). In each of these solutions, X also touches the outer boundary. (The cover and frontispiece of Clarke's book show a packing in which X doesn't touch the boundary, but it *doesn't* solve this problem: There's an edge where X meets I, and there's a point where X meets P.) There also are two packings in which the edges of X touch only F, N, U, and the boundary, but not V.

On the other hand, there are just 6 solutions if we allow only F, N, U, V to touch X's corner points. One of them, shown below, has X touching the short side and seems to match the quotation best. These 6 solutions can be found in just 47 Mμ, by introducing 60 secondary items as sort of an "upper level" to the board: All placements of X occupy the normal five lower-level cells, plus up to 16 upper-level cells that touch them; all placements of F, N, U, V are unchanged; all placements of the other seven pieces occupy both the lower and the upper level. This nicely forbids them from touching X.



256. (a) We could set this up as twelve separate exact cover problems, one for each pentomino omitted. But it's more interesting to consider all cases simultaneously, by giving a "free pass" to one pentomino as follows: Add a new primary item '`#`', and twelve new options '`# O`', '`# P`', ..., '`# Z`'. The sixty items ij are demoted to secondary status.

To remove symmetry, delete 3/4 of the options for piece V; also make its new option '`# V` s', and add 's' to 3/4 of the options for piece W, where 's' is a new secondary item. That makes a total of 1194 options, involving $13 + 61$ items.

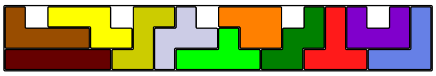
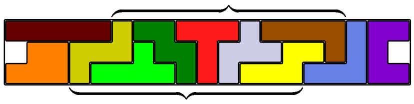
If Algorithm D branches first on `#`, the effect is equivalent to 12 separate runs; the search tree has 7.9 billion nodes, and the run time is 16.8 teramems. But if we modify the algorithm so that it branches on `#` only when necessary (see answer 83), the algorithm is able to save some time by making decisions that are common to several subcases. Its search tree then has 7.3 billion nodes, and the run time is 15.1 teramems. Of course both methods give the same answer, which is huge: 118,034,464.

(b) Now keep items ij primary, but introduce 60 new secondary items ij'. There are 60 new options '$ij\ ij'\ (i{+}1)j'\ i(j{+}1)'\ (i{+}1)(j{+}1)'$', where we omit items containing

Potts
Gardner
secondary items
symmetry

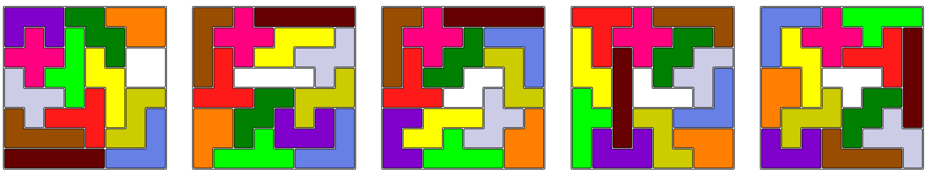
$(i{+}1)$ when $i = 2$ or $(j{+}1)$ when $j = 19$. This problem has 1254 options involving 73+61 items. Its search tree (with deprecated `#` branching) has about 950 million nodes; it finds 4,527,002 solutions, after about 1.5 teramems of computation.

A related, but much simpler, problem asks for packings in which exactly one hole appears in each of the column pairs {`1`, `2`}, {`5`, `6`}, {`9`, `a`}, {`d`, `e`}, {`h`, `i`}. That one has 1224 options, 78+1 items, 20 meganodes, 73 gigamems, and 23642 solutions. Here's one:

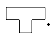
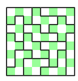
(c) A setup like the one in (a) yields 1127 options, 13 + 58 items, 1130 meganodes, 2683 gigamems, 22237 solutions. (One the noteworthy solutions is illustrated above.)

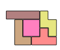
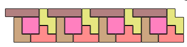
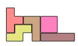
257. Restrict X to five essentially different positions; if X is on the diagonal, also keep Z unflipped by using the secondary item 's' as in answer 250. There are respectively (16146, 24600, 23619, 60608, 25943) solutions, found in (19.8, 35.4, 27.3, 66.6, 34.5) Gμ.



In each case the tetromino can be placed anywhere that doesn't immediately cut off a region of one or two squares. [The twelve pentominoes first appeared in print when H. E. Dudeney published *The Canterbury Puzzles* in 1907. His puzzle #74, "The Broken Chessboard," presented the first solution shown above, with pieces checkered in black and white. That parity restriction, with the further condition that no piece is turned over, would reduce the number of solutions to only 4, findable in 120 Mμ.]

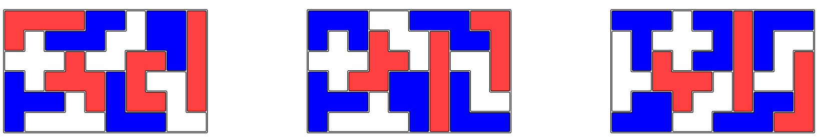
The 60-element subsets of the chessboard that *can't* be packed with the pentominoes have been characterized by M. Reid in *JRM* **26** (1994), 153–154.

258. Yes, in seven essentially different ways. To remove symmetry, we can make the O vertical and put the X in the right half. (The pentominoes will have a total of $6 \times 2 + 5 \times 3 + 4 = 31$ black squares; therefore the tetromino *must* be .) 

259. These shapes can't be packed in a rectangle. But we can use the "supertile"  to make an infinite strip $\cdots$  $\cdots$. We can also tile the plane with a supertile like , or even use a generalized torus such as  (see exercise 7–137). That supertile was used in 2009 by George Sicherman to make tetromino wallpaper.

260. The 2339 solutions contain 563 that satisfy the "tatami" condition: No four pieces meet at any one point. Each of those 563 leads to a simple 12-vertex graph coloring problem; for example, the SAT methods of Section 7.2.2.2 typically need at most two or three kilomems to decide each case.

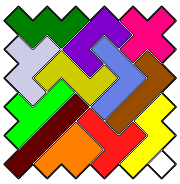
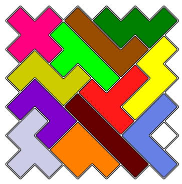
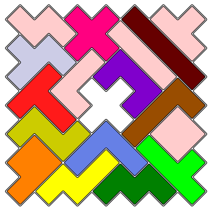
It turns out that exactly 94 are three-colorable, including the second solution to exercise 255(b). Here are the three for which W, X, Y, Z all have the same color:



flipped
Dudeney
parity
one-sided pentominoes
Reid
symmetry
torus
torus, generalized
Sicherman
wallpaper
tatami
SAT

262. Both shapes have 8-fold symmetry, so we can save a factor of nearly 8 by placing the X in (say) the north-northwest octant. If X thereby falls on the diagonal, or in the middle column, we can insist that the Z is not flipped, by introducing a secondary item 's' as in answer 252. Furthermore, if X occurs in dead center — this is possible only for shape (i) — we use 'c' as in that answer to prohibit also any rotation of the W.

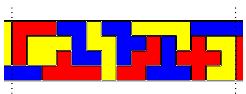
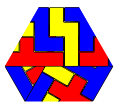
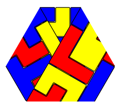
Thus find (a) 10 packings, in 3.5 Gμ; (b) 7302 packings, in 353 Gμ; for instance

 ,  ;  .

It turns out that the monomino must appear in or next to a corner, as shown. [The first solution to shape (i) with monomino in the corner was sent to Martin Gardner by H. Hawkins in 1958. The first solution of the other type was published by J. A. Lindon in *Recreational Mathematics Magazine* #6 (December 1961), 22. Shape (ii) was introduced and solved much earlier, by G. Fuhlendorf in *The Problemist: Fairy Chess Supplement* **2**, 17 and 18 (April and June, 1936), problem 2410.]

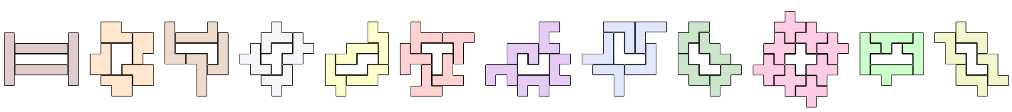
263. (Notice that width 3 would be impossible, because every faultfree placement of the V needs width 4 or more.) We can set up an exact cover problem for a 4×19 rectangle in the usual way; but then we make cell $(x, y + 15)$ identical to $(3 - x, y)$ for $0 \le x < 4$ and $0 \le y < 5$, essentially making a half-twist when the pattern begins to wrap around. There are 60 symmetries, and care is needed to remove them properly. The easiest way is to put X into a fixed position, and allow W to rotate at most 90°.

This exact cover problem has 850 solutions, 502 of which are faultfree. Here's one of the 29 strongly three-colorable ones, shown before and after its ends are joined:

 top:  bottom: 

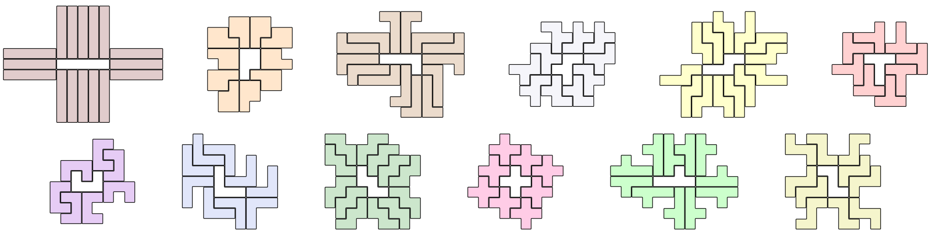
264. It's also possible to wrap *two* cubes of size $\sqrt{5}\times\sqrt{5}\times\sqrt{5}$, as shown by F. Hansson; see *Fairy Chess Review* **6** (1947–1948), problems 7124 and 7591. A full discussion appears in *FGbook*, pages 685–689.

265. It's easy to set up an exact cover problem in which the cells touching the polyomino are primary items, while other cells are secondary, and with options restricted to placements that contain at least one primary item. Postprocessing can then remove spurious solutions that contain holes. Typical answers for (a) are



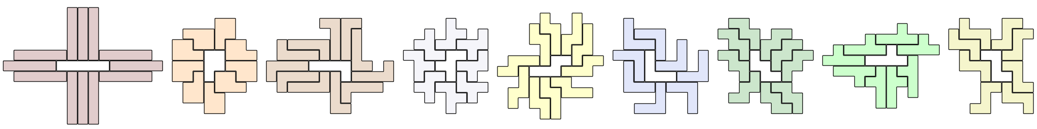
flipped
secondary item
Gardner
Hawkins
Lindon
Fuhlendorf
symmetries
three-colorable
Hansson

representing respectively (9, 2153, 37, 2, 17, 28, 18, 10, 9, 2, 4, 1) cases. For (b) they're



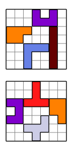
representing (16, 642, 1, 469, 551, 18, 24, 6, 4, 2, 162, 1). The total number of fences is respectively (3120, 1015033, 8660380, 284697, 1623023, 486, 150, 2914, 15707, 2, 456676, 2074), after weeding out respectively (0, 0, 16387236, 398495, 2503512, 665, 600, 11456, 0, 0, 449139, 5379) cases with holes. (See *MAA Focus* **36**, 3 (June/July 2016), 26; **36**, 4 (August/September 2016), 33.) Of course we can also make fences for one shape by using *other* shapes; for example, there's a beautiful way to fence a Z with 12 Ws, and a unique way to fence one pentomino with only *three* copies of another.

266. The small fences of answer 265(a) already meet this condition — except for the X, which has *no* tatami fence. The large fences for T and U in 165(b) are also good. But the other nine fences can no longer be as large:



[The tatami condition can be incorporated into the exact cover problem by using color controls: Introduce a secondary item for every potential edge between tiles, with values `t` and `f`. Also introduce a primary item p for every corner point; p will appear only in four options 'p e:`f`', one for each edge e that touches p. In every option for the placement of a piece, include the items 'e:`f`' for every edge *internal* to that piece, and 'e:`t`' for every edge at the *boundary* of that piece. Then every point will be next to a nonedge. However, for this exercise it's best simply to apply the tatami condition directly to each ordinary solution, before postprocessing for hole-removal.]

267. This problem is readily solved with the "second death" algorithm of exercise 18, by letting the four designated piece names be the *only* primary items. The answers to both (a) and (b) are unique. [See M. Gardner, *Scientific American* **213**, 4 (October 1965), 96–102, for Golomb's conjectures about minimum blocking configurations on larger boards.]

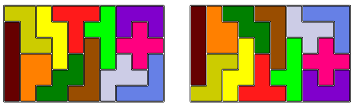
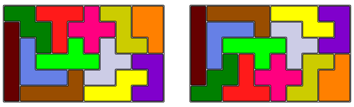


268. This exercise, with 3 × 30, 5 × 18, 6 × 15, and 9 × 10 rectangles, yields four increasingly difficult benchmarks for the exact cover problem, having respectively (46, 686628, 2562928, 10440433) solutions. Symmetry can be broken as in exercise 252. The 3 × 30 case was first resolved by J. Haselgrove; the 9 × 10 packings were first enumerated by A. Wassermann and P. Östergård, independently. [See *New Scientist* **12** (1962), 260–261; J. Meeus, *JRM* **6** (1973), 215–220; and *FGbook* pages 455, 468–469.] Algorithm D needs (.006, 5.234, 15.576, 63.386) teramems to find them.

269. Two solutions are now equivalent only when related by 180° rotation. Thus there are $2 \cdot 2339/64 = 73.09375$ solutions per problem, on average. The minimum (42) and

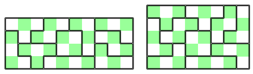
color controls
gadget
second death
Gardner
pentominoes, shortest games
benchmarks
Haselgrove
Wassermann
Östergård
Meeus
180° rotation
central symmetry

maximum (136) solution counts occur for the cases

(a)  ; (b)  .

[In *U.S. Patent 2900190* (1959, filed 1956), J. Pestiau remarked that these 64 problems would give his pentomino puzzle “unlimited life and utility.”]

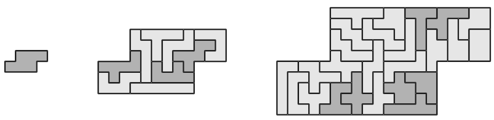
270. There are no ways to fill 2×20; 66×4 ways to fill 4×10; 84×4 ways to fill 5×8. None of the solutions are symmetrical. [See R. K. Guy, *Nabla* **7** (1960), 99–101.]



275. Most of the hexominoes will have three black cells and three white cells, in any “checkering” of the board. However, eleven of them (shown as darker gray in the illustration) will have a two-to-four split. Thus the total number of black cells will always be an even number between 94 and 116, inclusive. But a 210-cell rectangle always contains exactly 105 black cells. [See *The Problemist: Fairy Chess Supplement* **2**, 9–10 (1934–1935), 92, 104–105; *Fairy Chess Review* **3**, 4–5 (1937), problem 2622.]

Benjamin’s triangular shape, on the other hand, has $1+3+5+\cdots+19 = 10^2 = 100$ cells of one parity and $\binom{20}{2} - 10^2 = 110$ of the other. It can be packed with the 35 hexominoes in a huge number of ways, probably not feasible to count exactly.

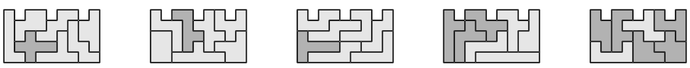
276. The parity considerations in answer 275 tell us that this is possible only for the “unbalanced” hexominoes, such as the one shown. And in fact, Algorithm D readily finds solutions for all eleven of those, too numerous to count. Here’s an example:



[See *Fairy Chess Review* **6** (April 1947) through **7** (June 1949), problems 7252, 7326, 7388, 7460, 7592, 7728, 7794, 7865, 7940, 7995, 8080. See also the similar problem 7092.]

277. Each castle must contain an odd number of the eleven unbalanced hexominoes (see answer 275). Thus we can begin by finding all sets of seven hexominoes that can be packed into a castle: This amounts to solving $\binom{11}{1} + \binom{11}{3} + \binom{11}{5} + \binom{11}{7} = 968$ exact cover problems, one for each potential choice of unbalanced elements. Each of those problems is fairly easy; the 24 balanced hexominoes provide secondary items, while the castle cells and the chosen unbalanced elements are primary. In this way we obtain 39411 suitable sets of seven hexominoes, with only a moderate amount of computation.

That gives us *another* exact cover problem, having 35 items and 39411 options. This secondary problem turns out to have exactly 1201 solutions (found in just 115 Gμ), each of which leads to at least one of the desired overall packings. Here’s one:



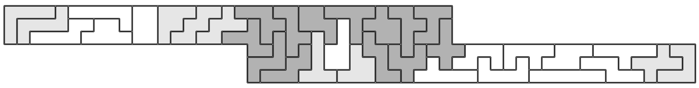
In this example, two of the hexominoes in the rightmost castle can be flipped vertically; and of course the entire contents of each castle can independently be flipped horizontally. Thus we get 64 packings from this particular partition of the hexominoes (or maybe $64 \times 5!$, by permuting the castles), but only two of them are “really” distinct. Taking multiplicities into account, there are 1803 “really” distinct packings altogether.

Patent
Pestiau
Guy
checkering
parity
parity
exact cover
factoring

[Frans Hansson found the first way to pack the hexominoes into five equal shapes, using  as the container; see *Fairy Chess Review* **8** (1952–1953), problem 9442. His container admits 123189 suitable sets of seven, and 9298602 partitions into five suitable sets instead of only 1201. Even more packings are possible with the container , which has 202289 suitable sets and 3767481163 partitions!]

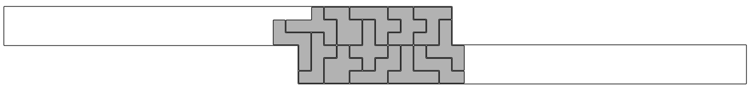
In 1965, M. J. Povah packed all of the hexominoes into containers of shape , using *seven* sets of *five*; see *The Games and Puzzles Journal* **2** (1996), 206.

278. By exercise 275, m must be odd, and less than 35. F. Hansson posed this question in *Fairy Chess Review* **7** (1950), problem 8556. He gave a solution for $m = 19$,

,

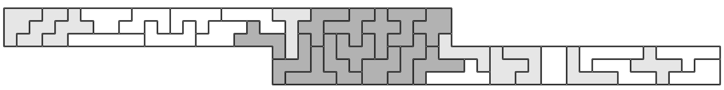
and claimed without proof that 19 is optimum. The 13 dark gray hexominoes in this diagram cannot be placed in either "arm"; so they must go in the center. (Medium gray indicates pieces that have parity restrictions in the arms.) Thus we cannot have $m \geq 25$.

When $m = 23$, there are 39 ways to place all of the hard hexominoes, such as

.

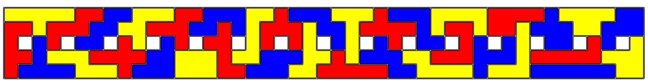
However, none of these is completable with the other 22; hence $m \leq 21$.

When $m = 21$, the hard hexominoes can be placed in 791792 ways, without creating a region whose size isn't a multiple of 6 and without creating more than one region that matches a particular hexomino. Those 791792 ways have 69507 essentially distinct "footprints" of occupied cells, and the vast majority of those footprints appear to be impossible to fill. But in 2016, George Sicherman found the remarkable packing

,

which not only solves $m = 21$, it yields solutions for $m = 19, 17, 15, 11, 9, 7, 5$, and 3 by simple modifications. Sicherman also found separate solutions for $m = 13$ and $m = 1$.

279. Stead's original solution makes a very pleasant three-colored design:



[See *Fairy Chess Review* **9** (1954), 2–4; also *FGbook*, pages 659–662.]

This problem is best solved via the techniques of dynamic programming (Section 7.7), *not* with Algorithm D, because numerous subproblems are equivalent.

281. Make options for the pentominoes in cells xy for $0 \leq x < 8$, $0 \leq y < 10$ as in exercise 240, and also for the tetrominoes in cells xy for $1 \leq x < 7$, $1 \leq y < 9$. In the latter options include also items xy':0 for all cells xy *in* the tetromino, as well as xy':1 for all other cells xy *touching* the tetromino, where the items xy' for $0 \leq x < 8$ and $0 \leq y < 10$ are secondary. We can also assume that the center of the X pentomino lies in the upper left corner. There are 168 solutions, found after 1.5 Tμ of computation. (Another way to keep the tetrominoes from touching would be to introduce secondary items for the *vertices* of the grid. Such items are more difficult to implement, however, because they behave differently under the rotations of answer 240.)

Hansson
Povah
Hansson
Sicherman
strongly three-colorable
dynamic programming

[Many problems that involve placing the tetrominoes and pentominoes together in a rectangle were explored by H. D. Benjamin and others in the *Fairy Chess Review*, beginning already with its predecessor *The Problemist: Fairy Chess Supplement* (1936), problem 2171. But this particular question seems to have been raised first by Michael Keller in *World Game Review* **9**, (1989), xx.]

282. At present, not a single solution to this puzzle is known, although intuition suggests that enormously many of them ought to be possible. P. J. Torbijn and J. Meeus [*JRM* **32** (2003), 78–79] have exhibited solutions for rectangles of sizes 6×45, 9×30, 10×27, and 15×18.

298. (a) Represent the tree as a sequence $a_0a_1 \ldots a_{2n+1}$ of nested parentheses; then $a_1 \ldots a_{2n}$ will represent the corresponding root-deleted forest, as in Algorithm 7.2.1.6P. The left boundary of the corresponding parallomino is obtained by mapping each '(' into N or E, according as it is immediately followed by '(' or ')'. The right boundary, similarly, maps each ')' into N or E according as it is immediately *preceded* by ')' or '('. For example, the parallomino for forest 7.2.1.6–(2) is shown below with part (d).

(b) This series $wxy + w^2(xy^2 + x^2y) + w^3(xy^3 + 2x^2y^2 + x^3y) + \cdots$ can be written $wxyH(w, wx, wy)$, where $H(w, x, y) = 1/(1 - x - y - G(w, x, y))$ generates a sequence of "atoms" corresponding to places x, y, G where the juxtaposed boundary paths have the respective forms $\genfrac{}{}{0pt}{}{\texttt{E}}{\texttt{E}}$, $\genfrac{}{}{0pt}{}{\texttt{N}}{\texttt{N}}$, or $\genfrac{}{}{0pt}{}{\texttt{N}}{\texttt{E}}\langle\text{inner}\rangle\genfrac{}{}{0pt}{}{\texttt{E}}{\texttt{N}}$. The area is thereby computed by diagonals between corresponding boundary points. (In the example from (a), the area is 1+1+1+1+2+2+2+2+2+2+2+2+2+1+1; there's an "outer" G, whose H is $xyxyGy$, and an "inner" G, whose H is $xyyxyxxyy$.) Thus we can write G as a continued fraction,

$$G(w,x,y) = wxy/(1-x-y-wxy/(1-wx-wy-w^3xy/(1-w^2x-w^2y-w^5xy/(\cdots)))).$$

[A completely different form is also possible, namely $G(w, x, y) = x\frac{J_1(w,x,y)}{J_0(w,x,y)}$, where

$$J_0(w,x,y) = \sum_{n=0}^{\infty} \frac{(-1)^n y^n w^{n(n+1)/2}}{(1-w)(1-w^2)\ldots(1-w^n)(1-xw)(1-xw^2)\ldots(1-xw^n)};$$
$$J_1(w,x,y) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1} y^n w^{n(n+1)/2}}{(1-w)(1-w^2)\ldots(1-w^{n-1})(1-xw)(1-xw^2)\ldots(1-xw^n)}.$$

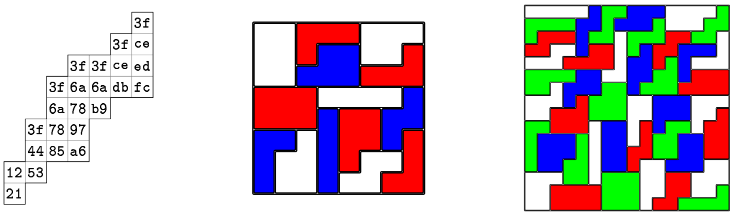
This form, derived via *horizontal* slices, disguises the symmetry between x and y.]

(c) Let $G(w, z) = G(w, z, z)$. We want $[z^n]\, G'(1, z)$, where differentiation is with respect to the first parameter. From the formulas in (b) we know that $G(1, z) = z(C(z) - 1)$, where $C(z) = (1 - \sqrt{1-4z})/(2z)$ generates the Catalan numbers. Partial derivatives $\partial/\partial w$ and $\partial/\partial z$ then give $G'(1, z) = z^2/(1-4z)$ and $G_\prime(1, z) = 1/\sqrt{1-4z} - 1$.

(d) This problem has four symmetries, because we can reflect about either diagonal. When $n = 5$, Algorithm D finds 801×4 solutions, of which 129×4 satisfy the tatami condition, and 16×4 are strongly three-colorable. (The tatami condition is easily enforced via secondary items in this case, because we need only stipulate that the upper right corner of one parallomino doesn't match the lower left corner of another.) When $n = 6$ there are oodles and oodles of solutions. All of the trees/parallominoes

Benjamin
Keller
Torbijn
Meeus
nested parentheses
forest
continued fraction
Bessel functions, gen'lized
Catalan numbers
tatami
strongly three-colorable
secondary items

thereby appear together in an attractive compact pattern.



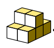
[*References:* D. A. Klarner and R. L. Rivest, *Discrete Math.* **8** (1974), 31–40; E. A. Bender, *Discrete Math.* **8** (1974), 219–226; I. P. Goulden and D. M. Jackson, *Combinatorial Enumeration* (New York: Wiley, 1983), exercise 5.5.2; M.-P. Delest and G. Viennot *Theoretical Comp. Sci.* **34** (1984), 169–206; W.-J. Woan, L. Shapiro, and D. G. Rogers, *AMM* **104** (1997), 926–931; P. Flajolet and R. Sedgewick, *Analytic Combinatorics* (Cambridge Univ. Press, 2009), 660–662.]

300. The same ideas apply, but with three coordinates instead of two, and with the elementary transformations $(x, y, z) \mapsto (y, x_{\max} - x, z)$, $(x, y, z) \mapsto (y, z, x)$.

Pieces (1, 2, . . . , 7) have respectively (12, 24, 12, 12, 12, 12, 8) base placements, leading to $144 + 144 + 72 + 72 + 96 + 96 + 64$ options for the $3 \times 3 \times 3$ problem.

302. It's tempting, but wrong, to try to compute the Somap by considering only the 240 solutions that have the tee in a fixed position and the claw restricted; the pairwise semidistances between these special solutions will miss many of the actual adjacencies. To decide if $u \,\text{—}\, v$, one must compare u to the 48 solutions equivalent to v.

(a) The strong Somap has vertex degrees $7^1\,6^7\,5^{19}\,4^{31}\,3^{59}\,2^{63}\,1^{45}\,0^{15}$; so an "average" solution has $(1 \cdot 7 + 7 \cdot 6 + \cdots + 15 \cdot 0)/240 \approx 2.57$ strong neighbors. (The unique vertex of degree 7 has the level-by-level structure $\begin{smallmatrix}333\\534\\552\end{smallmatrix}\ \begin{smallmatrix}114\\674\\652\end{smallmatrix}\ \begin{smallmatrix}174\\772\\662\end{smallmatrix}$ from bottom to top.)

The full Somap has vertex degrees $21^2\,18^1\,16^9\,15^{13}\,14^{10}\,13^{16}\,12^{17}\,11^{12}\,10^{16}\,9^{28}\,8^{26}\,7^{25}\,6^{26}\,5^{16}\,4^{17}\,3^3\,2^1\,1^1\,0^1$, giving an average degree ≈ 9.14. (Its unique isolated vertex is $\begin{smallmatrix}333\\432\\462\end{smallmatrix}\ \begin{smallmatrix}455\\466\\762\end{smallmatrix}\ \begin{smallmatrix}115\\715\\772\end{smallmatrix}$, and its only pendant vertex is $\begin{smallmatrix}333\\435\\222\end{smallmatrix}\ \begin{smallmatrix}755\\465\\462\end{smallmatrix}\ \begin{smallmatrix}771\\711\\466\end{smallmatrix}$. Two other noteworthy solutions, $\begin{smallmatrix}333\\436\\466\end{smallmatrix}\ \begin{smallmatrix}412\\452\\762\end{smallmatrix}\ \begin{smallmatrix}115\\755\\772\end{smallmatrix}$ and $\begin{smallmatrix}333\\436\\466\end{smallmatrix}\ \begin{smallmatrix}412\\455\\765\end{smallmatrix}\ \begin{smallmatrix}112\\752\\772\end{smallmatrix}$, are the only ones that contain the two-piece substructure .)

(b) The Somap has just two components, namely the isolated vertex and the 239 others. The latter has just three bicomponents, namely the pendant vertex, its neighbor, and the 237 others. Its diameter is 8 (or 21, if we use the edge lengths 2 and 3).

The strong Somap has a much sparser and more intricate structure. Besides the 15 isolated vertices, there are 25 components of sizes $\{8 \times 2, 6 \times 3, 4, 3 \times 5, 2 \times 6, 7, 8, 11, 16, 118\}$. Using the algorithm of Section 7.4.1, the large component breaks down into nine bicomponents (one of size 2, seven of size 1, the other of size 109); the 16-vertex component breaks into seven; and so on, totalling 58 bicomponents altogether.

[The Somap was first constructed by R. K. Guy, J. H. Conway, and M. J. T. Guy, without computer help. It appears on pages 910–913 of Berlekamp, Conway, and Guy's *Winning Ways*, where all of the strong links are shown, and where enough other links are given to establish near-connectedness. Each vertex in that illustration has been given a code name; for example, the five special solutions mentioned in part (a) have code names B5f, R7d, LR7g, YR3a, and R3c, respectively.]

303. "Factoring" with the residues $(i - j) \bmod 3$ and $(i + j) \bmod 3$, we see that the domino must go into adjacent cells with $(i - j) \bmod 3 \neq 1$ and $(i + j) \bmod 3 \neq 2$. That means either $\{(3i, 3j), (3i, 3j + 1)\}$ or $\{(3i + 1, 3j + 2), (3i + 2, 3j + 2)\}$. Conversely, it's easy to insert straight trominoes after placing a domino into any of those cell pairs.

geek art
Klarner
Rivest
Bender
Goulden
Jackson
Delest
Viennot
Woan
Shapiro
Rogers
Flajolet
Sedgewick
pendant vertex: of degree 1
diameter
Guy
Conway
Guy
Berlekamp
Conway
Guy
Factoring

304. Let the cubie coordinates be $51z$, $41z$, $31z$, $32z$, $33z$, $23z$, $13z$, $14z$, $15z$, for $z \in \{1,2,3\}$. Replace matrix A of the exact cover problem by a simplified matrix A' having only items $(1,2,3,4,5,6,7,\mathrm{S})$, where S is the sum of all items xyz of A where $x \cdot y \cdot z$ is odd. Any solution to A yields a solution to A' with item sums $(1,1,1,1,1,1,1,10)$. But that's impossible, because the S counts of pieces $(1, \ldots, 7)$ are at most (1, 2, 2, 1, 1, 1, 1). [See the Martin Gardner reference in answer 313.]

305. (a) The solution counts, ignoring symmetry reduction, are: 4×5 corral (2), gorilla (2), smile (2), 3×6 corral (4), face (4), lobster (4), castle (6), bench (16), bed (24), doorway (28), piggybank (80), five-seat bench (104), piano (128), shift 2 (132), 4×4 coop (266), shift 1 (284), bathtub (316), shift 0 (408), grand piano (526), tower 4 (552), tower 3 (924), canal (1176), tower 2 (1266), couch (1438), tower 1 (1520), stepping stones (2718). So the 4×5 corral, gorilla, and smile are tied for hardest, while stepping stones are the easiest. (The bathtub, canal, bed, and doorway each have four symmetries; the couch, stepping stones, tower 4, shift 0, bench, 4×4 coop, castle, five-seat bench, piggybank, lobster, piano, gorilla, face, and smile each have two. To get the number of *essentially distinct* solutions, divide by the number of symmetries.)

(b) Notice that the canal, bed, and doorway appear also in (a), as does the dryer (which is the same as "stepping stones"). The solution counts are: W-wall (0), almost W-wall (12), bed (24), apartments 2 (28), doorway (28), clip (40), tunnel (52), zigzag wall 2 (52), zigzag wall 1 (92), underpass (132), chair (260), stile (328), fish (332), apartments 1 (488), goldfish (608), canal (1176), steps (2346), dryer (2718); hence "almost W-wall" is the hardest of the possible shapes. Notice that the dryer, chair, steps, and zigzag wall 2 each have two symmetries, while the others in Fig. 80(b) all have four. The $3 \times 3 \times 3$ cube, with its 48 symmetries, probably is the easiest possible shape to make from the Soma pieces.

[Piet Hein himself published the tower 1, shift 2, stile, and zigzag wall 1 in his original patent; he also included the bathtub, bed, canal, castle, chair, steps, stile, stepping stones, shift 1, five-seat bench, tunnel, W-wall, and both apartments in his booklet for Parker Brothers. Parker Brothers distributed four issues of *The SOMA® Addict* in 1970 and 1971, giving credit for new constructions to Noble Carlson (fish, lobster), Mrs. C. L. Hall (clip, underpass), Gerald Hill (towers 2–4), Craig Kenworthy (goldfish), John W. M. Morgan (cot, face, gorilla, smile), Rick Murray (grand piano), and Dan Smiley (doorway, zigzag wall 2). Sivy Farhi published a booklet called *Somacubes* in 1977, containing the solutions to more than one hundred Soma cube problems including the bench, the couch, and the piggybank.]

306. By eliminating symmetries, there are (a) 421 distinct cases with cubies omitted on both layers, and (b) 129 with cubies omitted on only one layer. All are possible, except in the one case where the omitted cubies disconnect a corner cell. The easiest of type (a) omits $\{\mathtt{000}, \mathtt{001}, \mathtt{200}\}$ and has 3599 solutions; the hardest omits $\{\mathtt{100}, \mathtt{111}, \mathtt{120}\}$ and has 45×2 solutions. The easiest of type (b) omits $\{\mathtt{000}, \mathtt{040}, \mathtt{200}\}$ and has 3050 solutions; the hardest omits $\{\mathtt{100}, \mathtt{110}, \mathtt{140}\}$ and has 45×2 solutions. (The two examples illustrated have 821×2 and 68×4 solutions. Early Soma solvers seem to have overlooked them!)

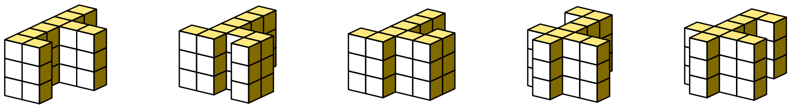
307. (a) The 60 distinct cases are all quite easy. The easiest has 3497 solutions and uses $\{\mathtt{002}, \mathtt{012}, \mathtt{102}\}$ on the top level; the hardest has 268 solutions and uses $\{\mathtt{002}, \mathtt{112}, \mathtt{202}\}$.

(b) Sixteen of the 60 possibilities are disconnected. Three of the others are also impossible — namely those that omit $\{\mathtt{01}z, \mathtt{13}z, \mathtt{21}z\}$ or $\{\mathtt{10}z, \mathtt{11}z, \mathtt{12}z\}$ or $\{\mathtt{10}z, \mathtt{11}z, \mathtt{13}z\}$. The easiest has 3554 solutions and omits $\{\mathtt{00}z, \mathtt{01}z, \mathtt{23}z\}$; the hardest of the possibles has only 8 solutions and omits $\{\mathtt{00}z, \mathtt{12}z, \mathtt{13}z\}$.

Gardner
symmetries
Hein
Parker Brothers
Carlson
Hall
Hill
Kenworthy
Morgan
Murray
Smiley
Farhi
symmetries

(The two examples illustrated have 132×2 and 270×2 solutions.)

308. All but 216 are realizable. Five cases have unique (1×2) solutions:



310. Every polycube has a minimum enclosing box for which it touches all six faces. If those box dimensions $a\times b\times c$ aren't too large, we can generate such polycubes uniformly at random in a simple way: First choose 27 of the abc possible cubies; try again if that choice doesn't touch all faces; otherwise try again if that choice isn't connected.

For example, when $a = b = c = 4$, about 99.98% of all choices will touch all faces, and about 0.1% of those will be connected. This means that about $.001\binom{64}{27} \approx 8 \times 10^{14}$ of the 27-cubie polycubes have a $4 \times 4 \times 4$ bounding box. Of these, about 5.8% can be built with the seven Soma pieces.

But most of the relevant polycubes have a larger bounding box; and in such cases the chance of solvability goes down. For example, $\approx 6.2\times 10^{18}$ cases have bounding box $4 \times 5 \times 5$; $\approx 3.3 \times 10^{18}$ cases have bounding box $3 \times 5 \times 7$; $\approx 1.5 \times 10^{17}$ cases have bounding box $2 \times 7 \times 7$; and only 1% or so of those cases are solvable.

Section 7.2.3 will discuss the enumeration of polycubes by their size.

312. Each interior position of the penthouse and pyramid that might or might not be occupied can be treated as a secondary item in the corresponding exact cover problem. We obtain 10×2 solutions for the staircase; $(223, 286) \times 8$ solutions for the penthouse with hole at the (bottom, middle); and 32×2 solutions for the pyramid, of which 2×2 have all three holes on the diagonal and 3×2 have no adjacent holes.

313. A full simulation of gravity would be quite complex, because pieces can be prevented from tipping with the help of their neighbors above and/or at their side. If we assume a reasonable coefficient of friction and an auxiliary weight at the top, it suffices to define stability by saying that a piece is stable if and only if at least one of its cubies is immediately above either the floor or a stable piece.

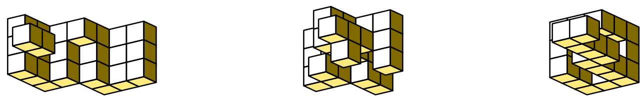
The given shapes can be packed in respectively 202×2, 21×2, 270×2, 223×8, and 122×2 ways, of which 202×2, 8×2, 53×2, 1×8, and 6×2 are stable. Going from the bottom level to the top, the layers $\substack{4\cdot\cdot7\\ \cdot\cdot\cdot\cdot\\ 3\cdot\cdot6}\ \substack{4477\\ 5567\\ 3366}\ \substack{54\cdot\cdot\\ 5\cdot\cdot\cdot\\ 31\cdot\cdot}\ \substack{22\cdot\cdot\\ 21\cdot\cdot\\ 21\cdot\cdot}$ give a decently stable cot; a fragile vulture comes from $\substack{2\cdot\cdot\cdot\\ \cdot\cdot\cdot7\\ 3\cdot\cdot\cdot}\ \substack{2447\\ 3177\\ 3566}\ \substack{2244\\ 1156\\ 3556}$; a delicate mushroom comes from $\substack{\cdot\cdot\cdot\\ \cdot7\cdot\\ \cdot\cdot\cdot}\ \substack{572\\ 577\\ 466}\ \substack{552\\ 3\cdot6\\ 446}\ \substack{322\\ 311\\ 341}$; and a delicate cantilever from $\substack{\cdot\cdot\cdot\\ 222\\ \cdot\cdot\cdot}\ \substack{\cdot\cdot\cdot\\ 255\\ \cdot\cdot\cdot}\ \substack{\cdot5\cdot\\ \cdot5\cdot\\ \cdot7\cdot}\ \substack{634\\ 674\\ 177}\ \substack{333\\ 664\\ 114}$. The author's cherished set of Skjøde Skjern Soma pieces, made of rosewood and purchased in 1967, includes a small square base that nicely stabilizes both mushroom and cantilever. The vulture needs a book on top.

[The casserole and cot are due respectively to W. A. Kustes and J. W. M. Morgan. The mushroom, which is hollow, is the same as B. L. Schwartz's "penthouse," but turned upside down; John Conway noticed that it then has a unique stable solution. See Martin Gardner, *Knotted Doughnuts* (1986), Chapter 3.]

314. Infinitely many cubies lie behind a wall; but it suffices to consider only the hidden ones whose distance is at most $27 - v$ from the v visible ones. For example, if the W-wall has coordinates as in answer 304, we have $v = 25$ and the two invisible cubies are $\{332, 331\}$. We're allowed to use any of $\{241, 242, 251, 252, 331, 332, 421, 422, 521, 522\}$ at distance 1, and $\{341, 342, 351, 352, 431, 432, 531, 532, 621, 622\}$ at distance 2. (The stated projection doesn't have left-right symmetry.) The X-wall is similar, but it has $v = 19$ and potentially (9, 7, 6, 3, 3, 2, 1) hidden cubies at distances 1 to 7 (omitting cases like 450, which is invisible at distance 2 but "below ground").

secondary item
author
Skjøde Skjern
Knutsen, see Skjøde Skjern
Kustes
Morgan
Schwartz
Conway
Gardner

Using secondary items for the optional cubies, we must examine each solution to the exact cover problem and reject those that are disconnected or violate the gravity constraint of exercise 313. Those ground rules yield 282 solutions for the W-wall, 612 for the X-wall, and a whopping 1,130,634 for the cube itself. (These solutions fill respectively 33, 275, and 13842 different sets of cubies.) Here are examples of some of the more exotic shapes that are possible, as seen from behind and below:



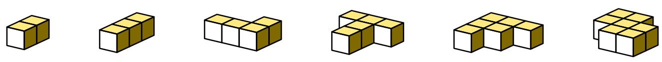
There also are ten surprising ways to make the cube façade if we allow hidden "underground" cubies: The remarkable construction $\begin{smallmatrix}\cdot\cdot\cdot\cdot\cdot\\ \cdot\cdot\cdot\cdot\cdot\\ \cdot\cdot\cdot 55\end{smallmatrix}\ \begin{smallmatrix}446\cdot\cdot\\ 7\cdot 65\cdot\\ 2\cdot\cdot 5\cdot\end{smallmatrix}\ \begin{smallmatrix}744\cdot\cdot\\ 7766\cdot\\ 2\cdot\cdot\cdot\cdot\end{smallmatrix}\ \begin{smallmatrix}333\cdot\cdot\\ 231\cdot\cdot\\ 211\cdot\cdot\end{smallmatrix}$ raises the entire cube one level *above* the floor, and is gravitationally stable, by exercise 313's criteria! Unfortunately, though, it falls apart — even with a heavy book on top.

[The false-front idea was pioneered by Jean Paul Francillon, whose construction of a fake W-wall was announced in *The SOMA® Addict* **2**, 1 (spring 1971).]

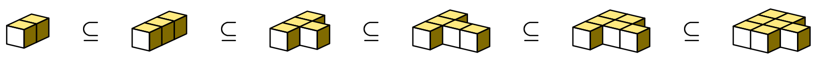
315. (a) Each of 13 solutions occurs in 48 equivalent arrangements. To remove the symmetry, place piece 7 horizontally, either (i) at the bottom or (ii) in the middle. In case (ii), add a secondary 's' item as in answer 250, and append 's' also to all placements of piece 6 that touch the bottom more than the top. Run time: 400 Kμ.

[This puzzle was number 39 in *Hoffmann's Puzzles Old and New* (1893). Another $3 \times 3 \times 3$ polycube dissection of historical importance, "Mikusinski's Cube," was described by Hugo Steinhaus in the 2nd edition of his *Mathematical Snapshots* (1950). That one consists of the ell and the two twist pieces of the Soma cube, plus the pentacubes B, C, and f of exercise 320; it has 24 symmetries and just two solutions.]

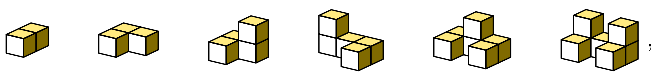
(b) Yes: Michael Reid, circa 1995, found the remarkable set



which also makes $9 \times 3 \times 1$ uniquely(!). George Sicherman carried out an exhaustive analysis of all relevant flat polyominoes in 2016, finding exactly 320 sets that are unique for $3 \times 3 \times 3$, of which 19 are unique also for $9 \times 3 \times 1$. In fact, one of those 19,



is the long-sought "Holy Grail" of $3 \times 3 \times 3$ cube decompositions: Its pieces not only have flatness and double uniqueness, they are nested (!!). There's also Yoshiya Shindo's



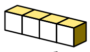
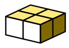
known as the "Neo Diabolical Cube" (1995); notice that it has 24 symmetries, not 48.

316. This piece can be modeled by a polycube with $20 + 20 + 27 + 3$ cubies, where we want to pack nine of them into a $9 \times 9 \times 9$ box. Divide that box into 540 primary cells (which must be filled) and 189 secondary cells (which will contain the 27 cubies of the simulated dowels). Answer 300 now yields an exact cover problem with 1536 options; and Algorithm D needs only 33 Mμ to discover 24 solutions, all equivalent by symmetry. (Or we could modify answer 300 so that all offsets have multiples of 3 in each coordinate; then there would be only 192 options, and the running time would go down to 8 Mμ.) One packing is $\begin{smallmatrix}122\\123\\443\end{smallmatrix}\ \begin{smallmatrix}567\\163\\849\end{smallmatrix}\ \begin{smallmatrix}557\\867\\899\end{smallmatrix}$, with dowels at $\begin{smallmatrix}010\\400\\030\end{smallmatrix}\ \begin{smallmatrix}070\\529\\080\end{smallmatrix}\ \begin{smallmatrix}000\\600\\000\end{smallmatrix}$.

gravitationally stable
Francillon
Hoffmann
Mikusinski's Cube
Steinhaus
pentacubes
Reid
Sicherman
Holy Grail
Shindo
Neo Diabolical Cube

One might be tempted to factor this problem, by first looking at all ways to pack nine solid bent trominoes into a $3 \times 3 \times 3$ box. That problem has 5328 solutions, found in about 5 Mμ; and after removing the 48 symmetries we're left with just 111 solutions, into which we can try to model the holes and dowels. But such a procedure is rather complicated, and it doesn't really save much time, if any.

Ronald Kint-Bruynseels, who designed this remarkable puzzle, also found that it's possible to drill holes in the solid cubies, parallel to the other two, without destroying the uniqueness of the solution(!). [*Cubism For Fun* **75** (2008), 16–19; **77** (2008), 13–18.]

317. The straight tetracube  and the square tetracube , together with the size-4 Soma pieces in (39), make a complete set.

We can fix the tee's position in the twin towers, saving a factor of 32; and each of the resulting 40 solutions has just one twist with the tee. Hence there are five inequivalent solutions, and 5×256 altogether.

The double claw has 63×6 solutions. But the cannon, with 1×4 solutions, can be formed in essentially only one way. (*Hint:* Both twists are in the barrel.)

There are no solutions to 'up 3'. But 'up 4' and 'up 5' each have 218×8 solutions (related by turning them upside down). Gravitationally, four of those 218 are stable for 'up 5'; the stable solution for 'up 4' is unique, and unrelated to those four.

References: Jean Meeus, *JRM* **6** (1973), 257–265; Nob Yoshigahara, *Puzzle World No. 1* (San Jose: Ishi Press International, 1992), 36–38.

318. All but 48 are realizable. The unique "hardest" realizable case, , has 2×2 solutions. The "easiest" case is the $2 \times 4 \times 4$ cuboid, with $11120 = 695 \times 16$ solutions.

320. (a) A, B, C, D, E, F, a, b, c, d, e, f, j, k, l, ..., z. (It's a little hard to see why reflection doesn't change piece 'l'. In fact, S. S. Besley once patented the pentacubes under the impression that there were 30 different kinds! See *U.S. Patent 3065970* (1962), where Figs. 22 and 23 illustrate the same piece in slight disguise.)

Historical notes: R. J. French, in *Fairy Chess Review* **4** (1940), problem 3930, was first to show that there are 23 different pentacube shapes, if mirror images are considered to be identical. The full count of 29 was established somewhat later by F. Hansson and others [*Fairy Chess Review* **6** (1948), 141–142]; Hansson also counted the $35 + 77 = 112$ mirror-inequivalent hexacubes. Complete counts of hexacubes (166) and heptacubes (1023) were first established soon afterwards by J. Niemann, A. W. Baillie, and R. J. French [*Fairy Chess Review* **7** (1948), 8, 16, 48].

(b) The cuboids $1\times3\times20$, $1\times4\times15$, $1\times5\times12$, and $1\times6\times10$ have of course already been considered. The $2 \times 3 \times 10$ and $2 \times 5 \times 6$ cuboids can be handled by restricting X to the bottom upper left, and sometimes also restricting Z, as in answers 250 and 252; we obtain 12 solutions (in 350 Mμ) and 264 solutions (in 2.5 Gμ), respectively.

The $3 \times 4 \times 5$ cuboid is more difficult. Without symmetry-breaking, we obtain 3940×8 solutions in about 200 Gμ. To do better, notice that X can appear in 11 essentially different positions: $(1+1^*)(1+1^*)$ in a 4×5 plane, 2^*+2^{**} in a 3×5 plane, and $2^* + 1^{**}$ in a 3×4 plane, where '*' denotes a case where symmetry needs to be broken down further because X is fixed by some symmetry. With 11 separate runs we can find $(923+558/2+402/2+376/4)+(1268/2+656/2+420/4+752/4)+(1480/2+720/2+352/4) = 3940$ solutions, in $4.9 + 3.3 + 3.1 + 2.4 + \cdots + 2.1 \approx 50$ Gμ.

[The fact that solid pentominoes will fill these cuboids was first demonstrated by D. Nixon and F. Hansson, *Fairy Chess Review* **6** (1948), problem 7560 and page 142.

factor
ell trominoes, see bent trominoes
bent trominoes
solid bent trominoes
L-cube puzzle
Kint-Bruynseels
straight tetracube
square tetracube
Meeus
Yoshigahara
cuboid
Besley
Patent
French
Hansson
hexacubes
heptacubes
Niemann
Baillie
cuboids
symmetry-breaking
Nixon
Hansson

Exact enumeration was first performed by C. J. Bouwkamp in 1967; see *J. Combinatorial Theory* **7** (1969), 278–280, and *Indagationes Math.* **81** (1978), 177–186.]

(c) Almost *any* subset of 25 pentacubes can probably do the job. But a particularly nice one is obtained if we simply omit o, q, s, and y, namely those that don't fit in a $3\times3\times3$ box. R. K. Guy proposed this subset in *Nabla* **7** (1960), 150, although he wasn't able to pack a $5 \times 5 \times 5$ at that time. The same idea occurred independently to J. E. Dorie, who trademarked the name "Dorian cube" [*U.S. Trademark 1,041,392* (1976)].

An amusing way to form such a cube is to make 5-level prisms in the shapes of the P, Q, R, U, and X pentominoes, using pieces $\{\mathrm{a,e,j,m,w}\}$, $\{\mathrm{f,k,l,p,r}\}$, $\{\mathrm{A,d,D,E,n}\}$, $\{\mathrm{c,C,F,u,v}\}$, $\{\mathrm{b,B,t,x,z}\}$; then use the packing in answer 251(!). This solution can be found with six very short runs of Algorithm D, taking only 300 megamems overall.

Another nice way, due to Torsten Sillke, is more symmetrical: There are 70,486 ways to partition the pieces into five sets of five that allow us to build an X-prism in the center (with piece x on top), surrounded by four P-prisms.

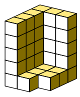
One can also assemble a Dorian cube from five cuboids, using one $1 \times 3 \times 5$, one $2 \times 2 \times 5$, and three $2 \times 3 \times 5$s. Indeed, there are zillions more ways, too many to count.

Bouwkamp
Guy
Dorie
Dorian cube
pentominoes
Sillke
partition

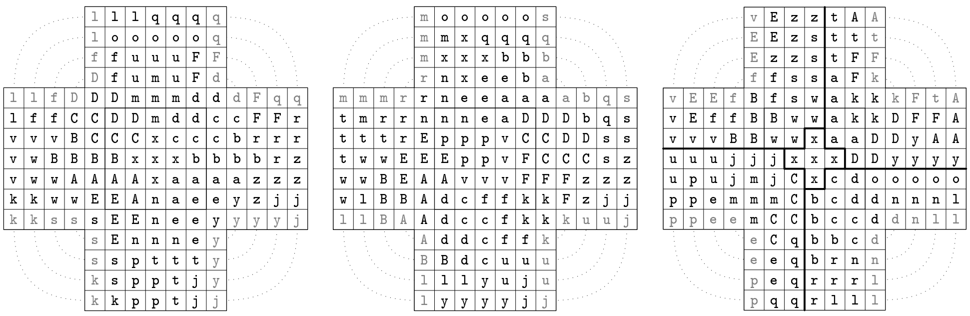
author
geek art
Sillke
Künzell
Farhi

321. (a) Make an exact cover problem in which a and A, b and B, ..., f and F are required to be in symmetrical position; there are respectively $(86, 112, 172, 112, 52, 26)$ placements for such 10-cubie "super-pieces." Furthermore, the author decided to force piece m to be in the middle of the top wall. Solutions were found immediately! So piece x was placed in the exact center, as an additional desirable constraint. Then there were exactly 20 solutions; the one below has also n, o, and u in mirror-symmetrical locations.

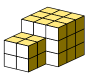
(b) The super-pieces now have $(59, 84, 120, 82, 42, 20)$ placements; the author also optimistically forced j, k, and m to be symmetrical about the diagonal, with m in the northwest corner. A long and apparently fruitless computation (34.3 teramems) ensued; but — hurrah — two closely related solutions were discovered at the last minute.

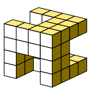


(c) This computation, due to Torsten Sillke [see *Cubism For Fun* **27** (1991), 15], goes much faster: The quarter-of-a-box shown here can be packed with seven non-x pentacubes in 55356 ways, found in 1.3 Gμ. As in answer 277, this yields a new exact cover problem, with 33412 different options. Then 11.8 Gμ more computation discovers seven suitable partitions into four sets of seven, one of which is illustrated here.



(a) (b) (c)





322. As in previous exercises, the key is to reduce the search space drastically, by asking for solutions of a special form. (Such solutions aren't unlikely, because pentacubes are so versatile.) Here we can break the given shape into four pieces: Three modules of size 3^3+2^3 to be packed with seven pentacubes, and one of size $4^3-3\cdot 2^3$ to be packed with eight pentacubes. The first problem has 13,587,963 solutions, found with 2.5 Tμ of computation; they involve 737,695 distinct sets of seven pentacubes. The larger problem has 15,840 solutions, found with 400 Mμ and involving 2075 sets of eight. Exactly covering those sets yields 1,132,127,589 suitable partitions; the first one found, $\{a, A, b, c, j, q, t, y\}$, $\{B, C, d, D, e, k, o\}$, $\{E, f, l, n, r, v, x\}$, $\{F, m, p, s, u, w, z\}$, works fine. (We need only one partition, so we needn't have computed more than a thousand or so solutions to the smaller problem.)

Pentacubes galore: Since the early 1970s, Ekkehard Künzell and Sivy Farhi have independently published booklets that contain hundreds of solved pentacube problems.

339. First we realize that every edge of the square must touch at least three pieces; hence the pieces must in fact form a 3×3 arrangement. Consequently any correct placement would also lead to a placement for nine pieces of sizes $(17-k)\times(20-k)$, ..., $(24-k)\times(25-k)$, into a $(65-3k)\times(65-3k)$ box. Unfortunately, however, if we try, say, $k=16$, Algorithm D quickly gives a contradiction.

X pentomino
tatami condition

But aha — a closer look shows that the pieces have *rounded corners*. Indeed, there's just enough room for pieces to get close enough together so that, if they truly were rectangles, they'd make a 1×1 overlap at a corner.

So we can take $k = 13$ and make nine pieces of sizes $4 \times 7, \ldots, 11 \times 12$, consisting of rectangles *minus* their corners. Those pieces can be packed into a 26×26 square, as if they were polyominoes (see exercise 240), but with the individual cells of the enclosing rectangle treated as secondary items because they needn't be covered. (Well, the eight cells adjacent to corners can be primary.) We can save a factor of 8 by insisting that the 9×11 piece appear in the upper left quarter, with its long side horizontal.

Algorithm D solves that problem in 620 gigamems — but it finds 43 solutions, most of which are unusable, because the missing corners give too much flexibility. The unique correct solution is easily identified, because a 1×1 overlap between rectangles in one place must be compensated by a 1×1 empty cell between rectangles in another. The resulting cross pattern (like the X pentomino) occurs in just one of the 43.

340. Let there be mn primary items p_{ij} for $0 \le i < m$ and $0 \le j < n$, one for each cell that should be covered exactly once. Also introduce m primary items x_i for $0 \le i < m$, as well as n primary items y_j for $0 \le j < n$. The exact cover problem has $\binom{m+1}{2} \cdot \binom{n+1}{2}$ options, one for each subrectangle $[a \,..\, b) \times [c \,..\, d)$ with $0 \le a < b \le m$ and $0 \le c < d \le n$. The option for that subrectangle contains $2 + (b-a)(d-c)$ items, namely x_a, y_c, and p_{ij} for $a \le i < b$, $c \le j < d$. The solutions correspond to reduced decompositions when we insist that each x_i be covered $[1 \,..\, n]$ times and that each y_j be covered $[1 \,..\, m]$ times. (We can save a little time by omitting x_0 and y_0.)

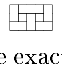
The 3×5 problem has 20165 solutions, found in 18 Mμ. They include respectively (1071, 3816, 5940, 5266, 2874, 976, 199, 22, 1) cases with (7, 8, …, 15) subrectangles.

341. The minimum is $m + n - 1$. Proof (by induction): The result is obvious when $m = 1$ or $n = 1$. Otherwise, given a decomposition into t subrectangles, $k \ge 1$ of them must be confined to the nth column. If two of those k are contiguous, we can combine them; the resulting dissection of order $t - 1$ reduces to either $(m-1) \times n$ or $m \times n$, hence $t - 1 \ge (m-1) + n - 1$. On the other hand if none of them are contiguous, the reduction of the first $n - 1$ columns is $m \times (n-1)$; hence $t \ge m + (n-1) - 1 + k$.

Close examination of this proof shows that a reduced decomposition has minimum order t if and only if its boundary edges form $m - 1$ horizontal lines and $n - 1$ vertical lines that don't cross each other. (In particular, the "tatami condition" is satisfied; see exercise 7.1.4–215.)

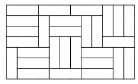
342. Simply remove the offending subrectangles, so that the cover problem has only $\bigl(\binom{m+1}{2}-1\bigr)\bigl(\binom{n+1}{2}-1\bigr)$ options. Now there are 13731 3×5 solutions, found in 11 Mμ, and (410, 1974, 3830, 3968, 2432, 900, 194, 22, 1) cases with (7, 8, …, 15) subrectangles.

343. Introduce additional primary items X_i for $0 < i < m$, to be covered $[1 \,..\, n-1]$ times, as well as Y_j for $0 < j < n$, to be covered $[1 \,..\, m-1]$ times. Then add items X_i for $a < i < b$ and Y_j for $c < j < d$ to the constraint for subrectangle $[a \,..\, b) \times [c \,..\, d)$.

Now the 3×5 problem has just 216 solutions, found in 1.9 megamems. They include (66, 106, 44) instances with (7, 8, 9) subrectangles. Just two of the solutions are symmetric under left-right reflection, namely  and its top-bottom reflection.

344. We can delete non-tromino options from the exact cover problem, thereby getting all faultfree tromino tilings that are reduced. If we also delete the constraints on x_i and y_j — and if we require X_i and Y_j to be covered $[1 \,..\, n]$ and $[1 \,..\, m]$ times instead of $[1 \,..\, n-1]$ and $[1 \,..\, m-1]$ — we obtain *all* of the $m \times n$ faultfree tromino tilings.

It is known that such nontrivial tilings exist if and only if $m, n \ge 7$ and mn is a multiple of 3. [See K. Scherer, *JRM* **13** (1980), 4–6; R. L. Graham, *The Mathematical Gardner* (1981), 120–126.] So we look at the smallest cases in order of mn: When $(m,n) = (7,9)$, $(8,9)$, $(9,9)$, $(7,12)$, $(9,10)$, we get respectively $(32,32)$, $(48,48)$, $(16,16)$, $(706,1026)$, $(1080,1336)$ solutions. Hence the assertion is false; a smallest counterexample is shown.



Scherer
Graham

347. Augment the exact cover problem of answer 342 by introducing $\binom{m+1}{2} + \binom{n+1}{2} - 2$ secondary items x_{ab} and y_{cd}, for $0 \le a < b \le m$ and $0 \le c < d \le n$, $(a,b) \ne (0,m)$, $(c,d) \ne (0,n)$. Include item x_{ab} and y_{cd} in the option for subrectangle $[a\,..\,b) \times [c\,..\,d)$. Furthermore, cover x_i $[1\,..\,m-i]$ times, not $[1\,..\,n]$; cover y_j $[1\,..\,n-j]$ times.

348. The hint follows because $[a\,..\,b) \times [0\,..\,d)$ cannot coexist motleywise with its left-right reflection $[a\,..\,b) \times [n{-}d\,..\,n)$. Thus we can forbid half of the solutions.

Consider, for example, the case $(m,n) = (7,7)$. Every solution will include x_{67} with some y_{cd}. If it's y_{46}, say, left-right reflection would produce an equivalent solution with y_{13}; therefore we disallow the option $(a,b,c,d) = (6,7,4,6)$. Similarly, we disallow $(a,b,c,d) = (6,7,c,d)$ whenever $7 - d < c$.

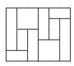
Reflection doesn't change the bottom-row rectangle when $c{+}d = 7$, so we haven't broken all the symmetry. But we can complete the job by looking also at the top-row rectangle, namely the option where x_{01} occurs with some $y_{c'd'}$. Let's introduce new secondary items t_1, t_2, t_3, and include t_c in the option that has x_{67} with $y_{c(7-c)}$. Then we include t_1, t_2, and t_3 in the option that has x_{01} with $y_{c'd'}$ for $c' + d' > 7$. We also add t_1 to the option with x_{01} and y_{25}; and we add both t_1 and t_2 to the option with x_{01} and y_{34}. This works beautifully, because no solution can have $c = c'$ and $d = d'$.

In general, we introduce new secondary items t_c for $1 \le c < n/2$, and we disallow all options $x_{(m-1)m}\ y_{cd}$ for which $c + d > n$. We put t_c into the option that contains $x_{(m-1)m}\ y_{c(n-c)}$. We put t_1 thru $t_{\lfloor (n-1)/2 \rfloor}$ into the option that contains $x_{01}\ y_{c'd'}$ when $c' {+} d' > n$. And we put t_1 thru $t_{c'-1}$ into the option that contains $x_{01}\ y_{c'(n-c')}$. (Think about it.)

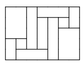
For example, when $m = n = 7$ there now are 717 options instead of 729, 57 secondary items instead of 54. We now find 352546 solutions after only 13.2 gigamems of computation, instead of 705092 solutions after 26.4. The search tree now has just 7.8 meganodes instead of 15.7.

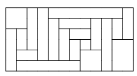
(It's tempting to believe that the same idea will break top-bottom symmetry too. But that would be fallacious: Once we've fixed attention on the bottommost row while breaking left-right symmetry, we've lost all symmetry between top and bottom.)

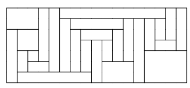
349. From any $m \times n$ dissection of order t we get two $(m+2) \times (n+2)$ dissections of order $t+4$, by enclosing it within two $1 \times m$ tiles and two $1 \times n$ tiles. So the claim follows by induction and the examples in exercise 347, together with a 5×6 example of order 10 — of which there are 8 symmetrical instances such as the one shown here. (This construction is faultfree, and it's also "tight": The order of every $m \times n$ dissection is at least $m+n-1$, by exercise 341.) faultfree



350. All 214 of the 5×7 motley dissections have order 11, which is far short of $\binom{6}{2} - 1 = 14$; and there are no 5×8s, 5×9s, or 5×10s. Surprisingly, however, 424 of the 696 dissections of size 6×12 do have the optimum order 20, and 7×17 dissections with the optimum order 27 also exist. Examples of these remarkable patterns are shown. (The case $m = 7$ is still not fully explored except for small n. For example, the total number of motley 7×17 dissections is unknown, and no 7×18s have yet been found. If we restrict attention to *symmetrical* dissections, the maximum orders for $5 \le m \le 8$ are 11 (5×7); 19 (6×11); 25 (7×15); 33 (8×21).)







352. The basic idea is to combine complementary options into a single option whenever possible. More precisely: (i) If $a+b=m$ and $c+d=n$, we retain the option as usual; it is self-complementary. (ii) Otherwise, if $a+b=m$ or $c+d=n$, reject the option; merging would be non-motley. (iii) Otherwise, if $a+b>m$, reject the option; we've already considered its complement. (iv) Otherwise, if $b=1$ and $c+d<n$, reject the option; its complement is illegal. (v) Otherwise, if $b>m/2$ and $c<n/2$ and $d>n/2$, reject the option; it intersects its complement. (vi) Otherwise merge the option with its complement. For example, when $(m,n)=(4,5)$, case (i) arises when $(a,b,c,d)=(1,3,2,3)$; the option is 'x_1 y_2 p_{12} p_{22} x_{13} y_{23}' as in answer 348. Case (ii) arises when $(a,b,c,d)=(1,3,0,1)$. Case (iii) arises when $(a,b)=(2,3)$. Case (iv) arises when $(a,b,c,d)=(0,1,0,1)$; the complement $(3,4,4,5)$ isn't a valid subrectangle in answer 348. Case (v) arises when $(a,b,c,d)=(1,3,1,3)$; cells p_{22} and p_{23} occur also in the complement $(1,3,2,4)$. And case (vi) arises when $(a,b,c,d)=(0,1,4,5)$; the merged option is the union of 'x_0 y_4 p_{04} x_{01} y_{45} t_1 t_2' and 'x_3 y_0 p_{30} x_{34} y_{01}'. (Well, x_0 and y_0 are actually omitted, as suggested in answer 340.)

Size 8×16 has (6703, 1984, 10132, 1621, 47) solutions, of orders (26, ..., 30).

353. (a) Again we merge compatible options, as in answer 352. But now $(a,b,c,d) \to (c,d,n-b,n-a) \to (n-b,n-c,n-b,n-a) \to (n-b,n-a,c,d)$, so we typically must merge *four* options instead of two. The rules are: Reject if $a=n-1$ and $c+d>n$, or $c=n-1$ and $a+b<n$, or $b=1$ and $c+d<n$, or $d=1$ and $a+b>n$. Also reject if (a,b,c,d) is lexicographically greater than any of its three successors. But accept, without merging, if $(a,b,c,d)=(c,d,n-b,n-a)$. Otherwise reject if $b>c$ and $b+d>n$, or if $b>n/2$ and $c<n/2$ and $d>n/2$, because of intersection. Also reject if $a+b=n$ or $c+d=n$, because of the motley condition. Otherwise merge four options into one.

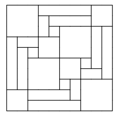
For example, the merged option when $n=4$ and $(a,b,c,d)=(0,1,2,4)$ is 'x_0 y_2 p_{02} p_{03} x_{01} y_{24} t_1 x_2 y_3 p_{23} p_{33} x_{24} y_{34} x_3 y_0 p_{30} p_{31} x_{34} y_{24} p_{00} p_{10} x_{02} y_{01}', except that x_0 and y_0 are omitted. Notice that it's important not to include an item x_i or y_j twice, when merging in cases that have $a=c$ or $b=d$ or $a=n-d$ or $b=n-c$.

(b) With bidirectional symmetry it's possible to have $(a,b,c,d)=(c,d,a,b)$ but $(a,b,c,d) \ne (n-d,n-c,n-b,n-a)$, or vice versa. Thus we'll sometimes merge two options, we'll sometimes merge four, and we'll sometimes accept without merging. In detail: Reject if $a=n-1$ and $c+d>n$, or $c=n-1$ and $a+b>n$, or $b=1$ and $c+d<n$, or $d=1$ and $a+b<n$. Also reject if (a,b,c,d) is lexicographically greater

than any of its three successors. But accept, without merging, if $a = c = n - d = n - b$. Otherwise reject if $b > c$ or $b > n - d$ or $a + b = n$ or $c + d = n$. Otherwise merge two or four distinct options into one.

Examples when $n = 4$ are: 'x_1 y_1 p_{11} p_{12} p_{21} p_{22} x_{13} y_{13}'; 'x_0 y_3 p_{03} x_{01} y_{34} t_1 x_3 y_0 p_{30} x_{34} y_{01}'; 'x_0 y_2 p_{02} x_{34} y_{23} t_1 x_1 y_3 p_{13} x_{12} y_{34} x_3 y_1 p_{31} x_{34} y_{12} x_2 y_0 p_{20} x_{23} y_{01}'; again with x_0 and y_0 suppressed.

pinwheel
factored
backtrack
van Hertog
Gardner
Cutler

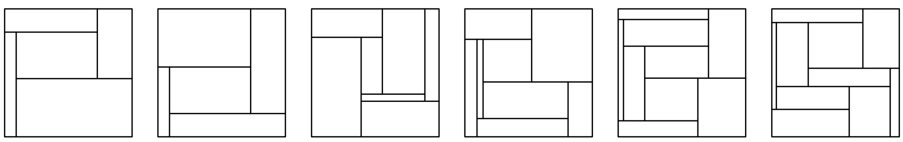


(c) The unique solution for $n = 10$ is shown. [The total number of such patterns for $n = (10, 11, \ldots, 16)$ turns out to be (1, 0, 3, 6, 28, 20, 354). All 354 of the 16×16 solutions are found in only 560 megamems; they have orders 34, 36, and 38–44. Furthermore the number of $n \times n$ motley dissections with symmetry (a), for $n = (3, 4, 5, \ldots, 16)$, turns out to be (1, 0, 2, 2, 8, 18, 66, 220, 1024, 4178, 21890, 102351, 598756, 3275503), respectively. Algorithm M needs 3.3 teramems when $n = 16$; those patterns have orders $4k$ and $4k + 1$ for $k = 8, 9, \ldots, 13$.]

355. The reduction of a perfectly decomposed rectangle is a motley dissection. Thus we can find all perfectly decomposed rectangles by "unreducing" all motley dissections.

For example, the only motley dissection of order 5 is the 3×3 pinwheel. Thus the perfectly decomposed $m \times n$ rectangles of order 5 with integer dimensions are the positive integer solutions to $x_1 + x_2 + x_3 = m$, $y_1 + y_2 + y_3 = n$ such that the ten values x_1, x_2, x_3, $x_1 + x_2$, $x_2 + x_3$, y_1, y_2, y_3, $y_1 + y_2$, $y_2 + y_3$ are distinct. Those equations are readily factored into two easy backtrack problems, one for m and one for n, each producing a list of five-element sets $\{x_1, x_2, x_3, x_1 + x_2, x_2 + x_3\}$; then we search for all pairs of disjoint solutions to the two subproblems. In this way we quickly see that the equations have just two essentially different solutions when $m = n = 11$, namely $(x_1, x_2, x_3) = (1, 7, 3)$ and $(y_1, y_2, y_3) = (2, 4, 5)$ or $(5, 4, 2)$. The smallest perfectly decomposed squares of order 5 therefore have size 11×11, and there are two of them (shown below); they were discovered by M. van Hertog, who reported them to Martin Gardner in May 1979. (Incidentally, a 12×12 square can also be perfectly decomposed.)

There are no solutions of order 6. Those of orders 7, 8, 9, 10 must come respectively from motley dissections of sizes 4×4, 4×5, 5×5, and 5×6. By looking at them all, we find that the smallest $n \times n$ squares respectively have $n = 18$, 21, 24, and 28. Each of the order-t solutions shown here uses rectangles of dimensions $\{1, 2, \ldots, 2t\}$, except in the case $t = 9$: There's a *unique* perfectly decomposed 24×24 square of order 9, and it uses the dimensions $\{1, 2, \ldots, 17, 19\}$.

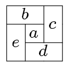


[W. H. Cutler introduced perfectly decomposed rectangles in *JRM* **12** (1979), 104–111.]

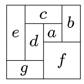
strict
NP-hard
linear programming

356. (a) False (but close). Let the individual dimensions be $z_1, \ldots, z_{2t}$, where $z_1 \le \cdots \le z_{2t}$. Then we have $\{w_1, h_1\} = \{z_1, z_{2t}\}$, $\{w_2, h_2\} = \{z_2, z_{2t-1}\}$, $\ldots$, $\{w_t, h_t\} = \{z_t, z_{t+1}\}$; consequently $z_1 < \cdots < z_t \le z_{t+1} < \cdots < z_{2t}$. But $z_t = z_{t+1}$ is possible.

(b) False (but close). If the reduced rectangle is $m \times n$, one of its subrectangles might be $1 \times n$ or $m \times 1$; a motley dissection must be *strict*.



(c) True. Label the rectangles $\{a, b, c, d, e\}$ as shown. Then there's a contradiction: $w_b > w_d \iff w_e > w_c \iff h_e < h_c \iff h_d < h_b \iff w_b < w_d$.



(d) The order can't be 6, because the reduction would then have to be a pinwheel together with a 1×3 subrectangle, and the argument in (c) would still apply. Thus the order must be 7, and we must show that the second dissection of exercise 347 doesn't work. Labeling its regions $\{a, \ldots, g\}$ as shown, we have $h_d > h_a$; hence $w_a > w_d$. Also $h_e > h_b$; so $w_b > w_e$. Oops: $w_f > w_g$ and $h_f > h_g$.

In the other motley 4×4 dissection of exercise 347 we obviously have

$$w_4 < w_5, \quad w_4 < w_6, \quad w_6 < w_7, \quad h_4 < h_3, \quad h_3 < h_1, \quad h_4 < h_2;$$

therefore $h_4 > h_5$, $h_4 > h_6$, $h_6 > h_7$, $w_4 > w_3$, $w_3 > w_1$, $w_4 > w_2$. Now $h_5 < h_6 \iff w_5 > w_6 \iff w_2 > w_3 \iff h_2 < h_3 \iff h_6 + h_7 < h_5$. Hence $h_5 < h_6$ implies $h_5 > h_6$; we must have $h_5 > h_6$, thus also $h_2 > h_3$. Finally $h_2 < h_1$, because $h_7 < h_5$.

(e) The condition is clearly necessary. Conversely, given any such pair of solutions, the rectangles $w_1 \times \alpha h_1, \ldots, w_t \times \alpha h_t$ are incomparable for all large enough α.

[Many questions remain unanswered: Is it NP-hard to determine whether or not a given motley dissection supports an incomparable dissection? Is there a motley dissection that supports incomparable dissections having two different permutation labels? Can a *symmetric* motley dissection ever support an incomparable dissection?]

357. (a) By exercise 356(d), the widths and heights must satisfy

$$\begin{aligned} &w_5 = w_2 + w_4, \quad && w_6 = w_3 + w_4, \quad && w_7 = w_1 + w_3 + w_4; \\ &h_3 = h_4 + h_5, && h_2 = h_4 + h_6 + h_7, && h_1 = h_4 + h_5 + h_6. \end{aligned}$$

To prove the hint, consider answer 356(a). Each z_j for $1 \le j \le t$ can be either w or h; then z_{2t+1-j} is the opposite. So there are 2^t ways to shuffle the w's and h's together.

For example, suppose all the h's come first, namely $h_7 < \cdots < h_1 \le w_1 < \cdots < w_7$:

$$\begin{aligned} 1 \le h_7,\ \ h_7 + 1 \le h_6,\ \ h_6 + 1 \le h_5,\ \ h_5 + 1 \le h_4,\ \ h_4 + 1 \le h_4 + h_5, \\ h_4 + h_5 + 1 \le h_4 + h_6 + h_7,\ \ h_4 + h_6 + h_7 + 1 \le h_4 + h_5 + h_6, \\ h_4 + h_5 + h_6 \le w_1,\ \ w_1 + 1 \le w_2,\ \ w_2 + 1 \le w_3,\ \ w_3 + 1 \le w_4, \\ w_4 + 1 \le w_2 + w_4,\ \ w_2 + w_4 + 1 \le w_3 + w_4,\ \ w_3 + w_4 + 1 \le w_1 + w_3 + w_4. \end{aligned}$$

The least perimeter in this case is the smallest value of $w_1 + w_2 + w_3 + w_4 + h_7 + h_6 + h_5 + h_4$, subject to those inequalities; and one easily sees that the minimum is 68, achieved when $h_7 = 2$, $h_6 = 3$, $h_5 = 4$, $h_4 = 5$, $w_1 = 12$, $w_2 = 13$, $w_3 = 14$, $w_4 = 15$.

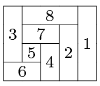
Consider also the alternating case, $w_1 < h_7 < w_2 < h_6 < w_3 < h_5 < w_4 \le h_4 < w_2 + w_4 < h_4 + h_5 < w_3 + w_4 < h_4 + h_6 + h_7 < w_1 + w_3 + w_4 < h_4 + h_5 + h_6$. This case turns out to be infeasible. (Indeed, any case with $h_6 < w_3 < h_5$ requires $h_4 + h_5 < w_3 + w_4$, hence it needs $h_4 < w_4$.) Only 52 of the 128 cases are actually feasible.

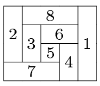
Each of the 128 subproblems is a classic example of linear programming, and a decent LP solver will resolve it almost instantly. The minimum perimeter with seven subrectangles is 35, obtained uniquely in the case $w_1 < w_2 < w_3 < h_7 < h_6 < h_5 < h_4 \le w_4 < w_5 < w_6 < w_7 < h_3 < h_2 < h_1$ (or the same case with $w_4 \leftrightarrow h_4$) by setting $w_1 = 1$, $w_2 = 2$, $w_3 = 3$, $h_7 = 4$, $h_6 = 5$, $h_5 = 6$, $h_4 = w_4 = 7$. The next-best case has perimeter 43. In one case the best-achievable perimeter is 103!

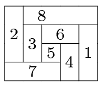
integer programming
Yao
Reingold
Sands
Jepsen
Nuij

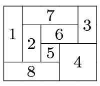
To find the smallest square, we simply add the constraint $w_1+w_2+w_3+w_4 = h_7+h_6+h_5+h_4$ to each subproblem. Now only four of the 128 are feasible. The minimum side, 34, occurs uniquely when $(w_1, w_2, w_3, w_4, h_7, h_6, h_5, h_4) = (3, 7, 10, 14, 6, 8, 9, 11)$.

(b) With eight subrectangles the reduced pattern is 4×5. We can place a 4×1 column at the right of either the 4×4 pattern or its transpose; or we can use one of the first two 4×5's in exercise 347. (The other six patterns can be ruled out, using arguments similar to those of answer 356.) The labeled diagrams are



,


,


,


.

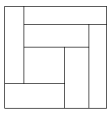
For each of these four choices there are 256 easy subproblems to consider. The best perimeters are respectively (44, 44, 44, 56); the best square sizes are respectively — and surprisingly — (27, 36, 35, 35). [With eight subrectangles we can dissect a significantly smaller square than we can with seven! Furthermore, no smaller square can be incomparably dissected, integerwise, because nine subrectangles would be too many.] One way to achieve perimeter 44 is with $(w_1, w_2, w_3, w_4, w_5, h_8, h_7, h_6, h_4) = (4, 5, 6, 7, 8, 1, 2, 3, 8)$ in the third diagram. The only way to achieve a square of side 27 is with $(w_1, w_2, w_3, w_4, w_5, h_8, h_7, h_6, h_4) = (1, 3, 5, 7, 11, 4, 6, 8, 9)$ in the first diagram.

These linear programs usually have integer solutions; but sometimes they don't. For example, the optimum perimeter for the second diagram in the case $h_8 < h_7 < w_1 < h_6 < w_2 < w_3 < w_4 < h_5$ turns out to be 97/2, achievable when $(w_1, w_2, w_3, w_4, w_5, h_8, h_7, h_6, h_4) = (7, 11, 13, 15, 17, 3, 5, 9, 17)/2$. The minimum rises to 52, if we restrict to integer solutions, achieved by $(w_1, w_2, w_3, w_4, w_5, h_8, h_7, h_6, h_4) = (4, 6, 7, 8, 9, 1, 3, 5, 9)$.

[The theory of incomparable dissections was developed by A. C. C. Yao, E. M. Reingold, and B. Sands in *JRM* **8** (1976), 112–119. For generalizations to three dimensions, see C. H. Jepsen, *Mathematics Magazine* **59** (1986), 283–292.]

358. This is an incomparable dissection in which exercise 356(d) applies. Let's try first to solve the equations $a(x+y+z) = bx = c(w+x) = d(w+x+y) = (a+b)w = (b+c)y = (b+c+d)z = 1$, by setting $b = x = 1$. We find successively $c = 1/(w+1)$, $a = (1-w)/w$, $y = (w+1)/(w+2)$, $d = (w+1)/(w(w+2))$, $z = (w+1)(w+3)/((w+2)(w+4))$. Therefore $x+y+z-1/a = (2w+3)(2w^2+6w-5)/((w-1)(w+2)(w+4))$, and we must have $2w^2+6w = 5$. The positive root of this quadratic is $w = (\surd-3)/2$, where $\surd = \sqrt{19}$.

Having decomposed the rectangle $(a+b+c+d)\times(w+x+y+z)$ into seven different rectangles of area 1, we normalize it, dividing (a, b, c, d) by $a+b+c+d = \frac{7}{15}(\surd+1)$ and dividing (w, x, y, z) by $w+x+y+z = \frac{5}{6}(\surd-1)$. This gives the desired tiling (shown), with rectangles of dimensions $\frac{1}{14}(7-\surd)\times\frac{1}{15}(7+\surd)$, $\frac{5}{42}(-1+\surd)\times\frac{1}{15}(1+\surd)$, $\frac{5}{21}\times\frac{3}{5}$, $\frac{1}{21}(8-\surd)\times\frac{1}{15}(8+\surd)$, $\frac{1}{21}(8+\surd)\times\frac{1}{15}(8-\surd)$, $\frac{5}{42}(1+\surd)\times\frac{1}{15}(-1+\surd)$, $\frac{1}{14}(7+\surd)\times\frac{1}{15}(7-\surd)$.



[See W. A. A. Nuij, *AMM* **81** (1974), 665–666. To get eight different rectangles of area 1/8, we can shrink one dimension by 7/8 and attach a rectangle $(1/8)\times 1$. Then to get nine of area 1/9, we can shrink the *other* dimension by 8/9 and attach a $(1/9)\times 1$ sliver. And so on. The eight-rectangle problem also has two other solutions, supported by the third and fourth 4×5 patterns in exercise 357(b).]

360. Let the back corner in the illustration be the point 777, and write just '*abcdef*' instead of $[a\,..\,b)\times[c\,..\,d)\times[e\,..\,f)$. The subcuboids are 670517 (270601) 176705 (012706) 051767 (060127), 561547 (260312) 475615 (122603) 154756 (031226), 351446 (361324) 463514 (243613) 144635 (132436), 575757 (020202), 454545 (232323) — with the 11 mirror images in parentheses — plus the central cubie 343434. Notice that each of the 28 possible intervals is used in each dimension, except $[0\,..\,4)$, $[1\,..\,6)$, $[2\,..\,5)$, $[3\,..\,7)$, $[0\,..\,7)$.

Hilbert
KIM
Gardner
puzzle
23 and me
author
author

I started from a central cube and built outwards, all the while staring at the 24-cell in Hilbert's Geometry and the Imagination.

— SCOTT KIM, letter to Martin Gardner (December 1975)

361. One solution is obtained by using the 7-tuples $(1,2,5,44,9,43,4)$, $(6,15,10,23,22,19,13)$, $(14,12,16,11,18,17,20)$, to "unreduce" the 1st, 2nd, 3rd coordinates. For example, subcuboid 670617 becomes $4\times(6{+}15{+}10{+}23{+}22)\times(12{+}16{+}11{+}18{+}17{+}20)$. The resulting dissection, into blocks of sizes $1\times87\times88$, $2\times42\times74$, $3\times21\times26$, $4\times76\times94$, $5\times10\times16$, $6\times82\times104$, $7\times33\times46$, $8\times15\times62$, $9\times18\times22$, $11\times23\times44$, $12\times31\times101$, $13\times71\times107$, $14\times95\times105$, $17\times54\times60$, $19\times56\times57$, $20\times61\times102$, $25\times27\times96$, $28\times49\times64$, $29\times41\times51$, $32\times37\times47$, $35\times48\times53$, $39\times45\times52$, $43\times55\times70$, makes a fiendishly difficult puzzle.

How were those magic 7-tuples discovered? An exhaustive search such as that of exercise 356 was out of the question. The author first looked for 7-tuples that would not lead to any dimensions in the "popular" ranges $[11\,..\,25]$ and $[30\,..\,42]$; there are 1130, of which the fourth was $(1,2,5,44,9,43,4)$. A subsequent search found 25112 7-tuples that don't conflict with this one, in the 23 relevant places; and those 7-tuples included 26 that don't conflict with each other.

(The cube size 108 can probably be decreased, but probably not by much.)

362. The exact cover problem of answer 347 is readily extended to 3D: The option for every admissible subcuboid $[a\,..\,b)\times[c\,..\,d)\times[e\,..\,f)$ has $6+(b-a)(d-c)(f-e)$ items, namely x_a y_c z_e x_{ab} y_{cd} z_{ef} and the cells p_{ijk} that are covered.

We can do somewhat better, as in exercise 348: Most of the improvement in that answer can be achieved also 3Dwise, if we simply omit cases where $a=l-1$ and either $c+d>m$ or $e+f>n$. Furthermore, if $m=n$ we can omit cases with $(e,f)<(c,d)$.

Without those omissions, Algorithm M handles the case $l=m=n=7$ in 98 teramems, producing 2432 solutions. With them, the running time is reduced to 43 teramems, and 397 solutions are found.

(The $7\times7\times7$ problem can be factored into subproblems, based on the patterns that appear on the cube's six visible faces. These patterns reduce to 5×5 pinwheels, and it takes only about 40 Mμ to discover all 152 possibilities. Furthermore, those possibilities reduce to only 5 cases, under the 48 symmetries of a cube. Each of those cases can then be solved by embedding the 5×5 reduced patterns into 7×7 unreduced patterns, considering $15^3=3375$ possibilities for the three faces adjacent to vertex 000. Most of those possibilities are immediately ruled out. Hence each of the five cases can be solved by Algorithm C in about 70 Gμ — making the total running time about 350 Gμ. However, this 120-fold increase in speed cost the author two man-days of work.)

All three methods showed that, up to isomorphism, exactly 56 distinct motley cubes of size $7\times7\times7$ are possible. Each of those 56 dissections has exactly 23 cuboids. Nine of them are symmetric under the mapping $xyz\mapsto(7-x)(7-y)(7-z)$; and one of those nine, namely the one in exercise 360, has six automorphisms.

[These runs confirm and slightly extend the work of W. H. Cutler in *JRM* **12** (1979), 104–111. His computer program found exactly 56 distinct possibilities, when restricting the search to solutions that have exactly 23 cuboids.]

Cutler
author

363. The author has confirmed this conjecture only for $l, m, n \le 8$ and $l+m+n \le 22$.

999. ...

INDEX AND GLOSSARY

Pope
Homer
WHEATLEY

There is a curious poetical index to the Iliad in Pope's Homer,
referring to all the places in which similes are used.
— HENRY B. WHEATLEY, *What is an Index?* (1878)

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.

Barris, Harry, 1.
DIMACS: *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, inaugurated in 1990.
Fields, Dorothy, 1.
MPR: Mathematical Preliminaries Redux, v.
Short, Robert Allen, iii.

Nothing else is indexed yet (sorry).

Preliminary notes for indexing appear in the upper right corner of most pages.

If I've mentioned somebody's name and forgotten to make such an index note, it's an error (worth \$2.56).