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

May 24, 2016

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.

Zeroth printing (revision -91), 24 May 2016

May 24, 2016

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 those carefully-checked volumes, alas, were subsequently found to contain thousands of mistakes.

Given this caveat, I hope that my errors this time will not be so numerous and/or obtrusive that you will be discouraged from reading the material carefully. I did try to make 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 . . . ; I've also implicitly mentioned or posed additional unsolved questions in the answers to exercises 81, ... 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. Knuth

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 20, 21, 40, ... Furthermore I've credited exercises ... to unpublished work of .... Have any of those results ever appeared in print, to your knowledge?

* * *

Special thanks are due to ... for their detailed comments on my early attempts at exposition, as well as to 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:–)

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 2015

BARRIS
FIELDS
exact covering–
0s and 1s

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. Blah blah de blah blah blah.

* * *

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

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}. \tag{20}$$

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.

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." ...

* 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.

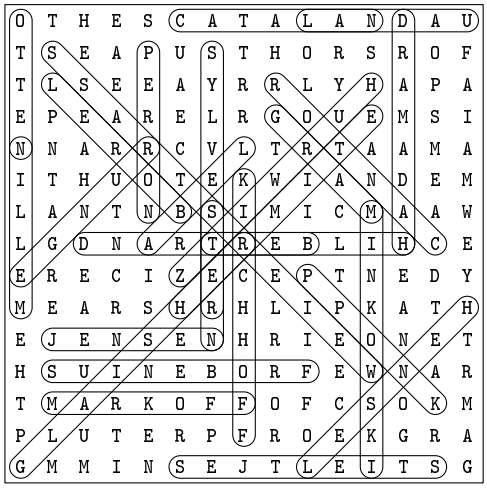


Fig. 72. Solution to the puzzle of the hidden mathematicians (Fig. 71). Notice that the central letter `R` actually participates in six different names:

`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*.

EXERCISES

NP-hard
unique solution
instant insanity
grope
binary operation
multiplication tables
idempotent
commutative
identity element
queen graph
colored
exact cover problem
bitwise manipulation
breadth-first
0s and 1s

19. [*M16*] Given an exact cover problem 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.]

20. [*M25*] Given an exact cover problem A, 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.

21. [*M21*] Continuing exercise 20, construct A' having *exactly* three 1s per column.

▸ **24.** [*30*] Given an $m \times n$ exact cover problem A with exactly three 1s per column, 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).)

▸ **26.** [*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 $\{0, 1, 2, 3\}$?

$$\begin{array}{cccccc} 0123 & 0321 & 0132 & 0231 & 0312 \\ 1032 & 3210 & 1023 & 3102 & 2130 \\ 2301 & 2103 & 3210 & 1320 & 3021 \\ 3210 & 1032 & 2301 & 2013 & 1203 \end{array}.$$

(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 columns (x, y) for $0 \le x, y < n$ and the following $n + (n^3 - n)/3$ rows:

i) $\{(x, x)\}$, for $0 \le x < n$;
ii) $\{(x, x), (x, y), (y, x)\}$, for $0 \le x < y < n$;
iii) $\{(x, y), (y, z), (z, x)\}$, 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).

27. [*21*] Modify the exact cover problem of exercise 26(d) in order to find the multiplication tables of (a) all idempotent gropes — gropes such that $x \circ x = x$ for all x; (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.

39. [*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.

40. [*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?

▸ **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 columns, 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 columns.

primary columns
bitwise AND
bitwise OR
nonprimary columns
n queens problem
word search puzzle
presidents
I'm not sure how many of these names should go in the index
connected
polyomino sudoku
sudoku

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 $\sum_{j=1}^{N} c_j[s_j \mathbin{\&} p = p]$.

N3. [Append r_k where possible.] Set $t \leftarrow r_k$. For $N \geq j \geq 1$, if $s_j \mathbin{\&} t = 0$, 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 $u_k \leftarrow r_k \mathbin{\&} \bar{f}_k$, 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 item from the database for which s_j does not contain $u_k \mathbin{\&} p$. We can also exploit the nonprimary columns 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.

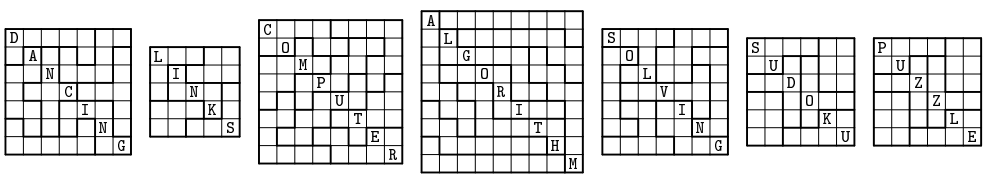
80. [*22*] Using the "word search puzzle" conventions of Figs. 71 and 72, show that the words ONE, TWO, THREE, FOUR, FIVE, SIX, SEVEN, EIGHT, NINE, TEN, ELEVEN, and TWELVE can all be packed into a 6×6 square, leaving one cell untouched.

▸ **81.** [*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?

90. [*24*] Find the unique solutions to the following examples of polyomino sudoku:



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

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)

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

SECTION 7.2.2.1

19. (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.

20. (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!

21. 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.

24. 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.]

26. (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 row of type (i) in an exact covering, define $x \circ x = x$; for each row of type (ii), define $x \circ x = y$, $x \circ y = y \circ x = x$; for each row 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 columns with k rows of size 1, all other rows 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*.

isomorphic
isomorphic
Bennett
Caltech
author
groups
quasigroup
semisymmetric quasigroup

27. (a) Eliminate the n columns for (x, x); use only the $2\binom{n}{3}$ rows of type (iii) for which $y \ne z$. (Idempotent gropes are equivalent to "Mendelsohn triples," which are families of $n(n-1)/3$ 3-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 $\binom{n+1}{2}$ columns (x, y) for $0 \le x \le y < n$; replace rows of type (ii) by $\{(x, x), (x, y)\}$ and $\{(x, y), (y, y)\}$ for $0 \le x < y < n$; replace those of type (iii) by $\{(x, y), (x, z), (y, z)\}$ 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 columns for which $x = 0$ or $y = 0$. Use only the $2\binom{n-1}{3}$ rows of type (iii) for $1 \le x < y, x < n$ and $y \ne z$. (Indeed, such systems are equivalent to idempotent gropes on the elements $\{1, \ldots, n-1\}$.)

Mendelsohn triples
Schröder
totally symmetric quasigroups
Stein
Steiner triple systems

39. Each of the 92 solutions to the eight queens problem (see Fig. 68) occupies eight of the 64 cells; so we must find eight disjoint solutions. Only 1897 updates of Algorithm D are needed 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. 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.]

eight queens problem
Gosset
Dudeney

12345678
78563412
46718235
23854167
84236751
51672384
67481523
512784

40. This is an exact cover problem with $92 + 312 + 396 + \cdots + 312 = 3284$ rows (see exercise 7.2.2–5). Algorithm D needs about 2 million updates to find the solution shown, and about 83 billion to find all 11,092 of them.

07348652
18650437
75421860
26835071
34072186
52183704
80564213
61207345

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

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

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 $N \leq 1$. 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 the exact cover problem in ($\star\star$), 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 $3n$ columns.]

frontier
n queens
Rivin
Zabih
Lamping
author

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 ($\star\star$) 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 a lot better: It needed only about 780K updates to find all $Q(11) = 2680$ solutions.

triply linked
binary search tree
backtracking algorithm
asymptotically
theory vs practice
practice vs theory

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 7.2.2–14). But in practice, a breadth-first approach needs too much space.

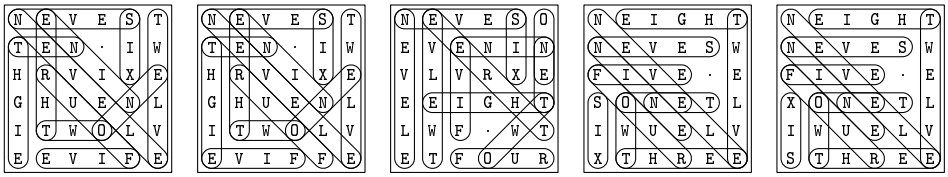
On the other hand, this method did beat Algorithm D on the n queen bees problem of exercise 7.2.2–15: When $n = 11$ its database grew to 364,864 items; it computed $H(11) = 596{,}483$ in just 30 Mμ, while Algorithm D needed 27 mega-updates.

n queen bees

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.]

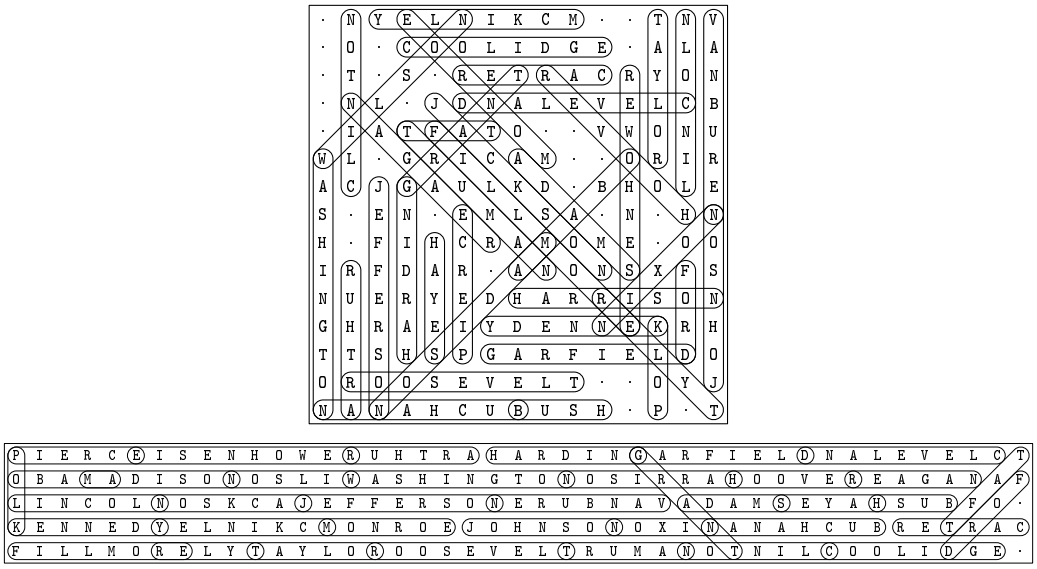
80. There are just five solutions; the latter two are flawed by being disconnected:

disconnected
Ocón de Oro
Gibat
author
interactive method
Gordon
Eckler
author



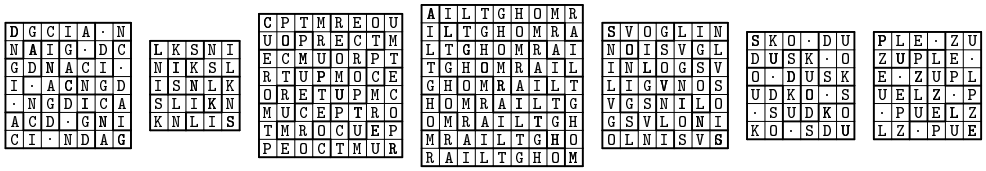
Historical note: Word search puzzles were invented in Spain by Pedro Ocón de Oro [I'm trying to learn the date], and independently in America by Norman E. Gibat in 1968.

81. (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.]

90. (The author designed these puzzles with the aid of exercises ??–??.)



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.

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).