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 8A

HAMILTONIAN PATHS AND CYCLES

DONALD E. KNUTH *Stanford University*

ADDISON–WESLEY

July 1, 2025

Internet
Stanford GraphBase
`MMIX`

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

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

See also `https://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 -69), 30 June 2025

BARRY
Internet

PREFACE

But that is not my point.
I have no point.
— DAVE BARRY (2002)

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, 4A, and 4B 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 8 previews Section 7.2.2.4 of *The Art of Computer Programming*, entitled "Hamiltonian paths and cycles." I haven't had time to write much of it yet — not even this preface!

* * *

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 65, ... 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 11, 12, 36, 37, 41, 42, 53, 55, 62, 63, 65, 71, 73, 84, 100, 106, 150, 151, 199, 200, 250, 260, 261, ... Furthermore I’ve credited exercises 79, . . . to unpublished work of Nikolai Beluhov and . . . . Have any of those results ever appeared in print, to your knowledge?

While writing this section I also wrote numerous programs for my own edification. (I usually can’t understand things well until I’ve tried to explain them to a machine.) Most of those programs were quite short, of course; but several of them are rather substantial, and possibly of interest to others. Therefore I’ve made a selection available by listing some of them on the following webpage:

`https://cs.stanford.edu/~knuth/programs.html`

In particular, prototypes of the main algorithms can be found there: Algorithm F (HAM-EULER); Algorithm H (SSHAM); Algorithm W (BACK-WARNSDORF). A few other programs are mentioned in the answers to certain exercises. If you want to see a program called FOO, look for FOO on that webpage. See also

`https://cs.stanford.edu/~knuth/programs/ham-benchmarks.tgz`

for the benchmark graphs in Table 1.

* * *

Special thanks are due to George Jelliss, Arnaud Lefebvre, Filip Stappers, Udo Wermuth, . . . 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
99 Umbruary 2016

D. E. K.

I have twenty years’ work ahead of me
to finish The Art of Computer Programming.
— DONALD E. KNUTH, letter to John Ewing (04 September 1990)

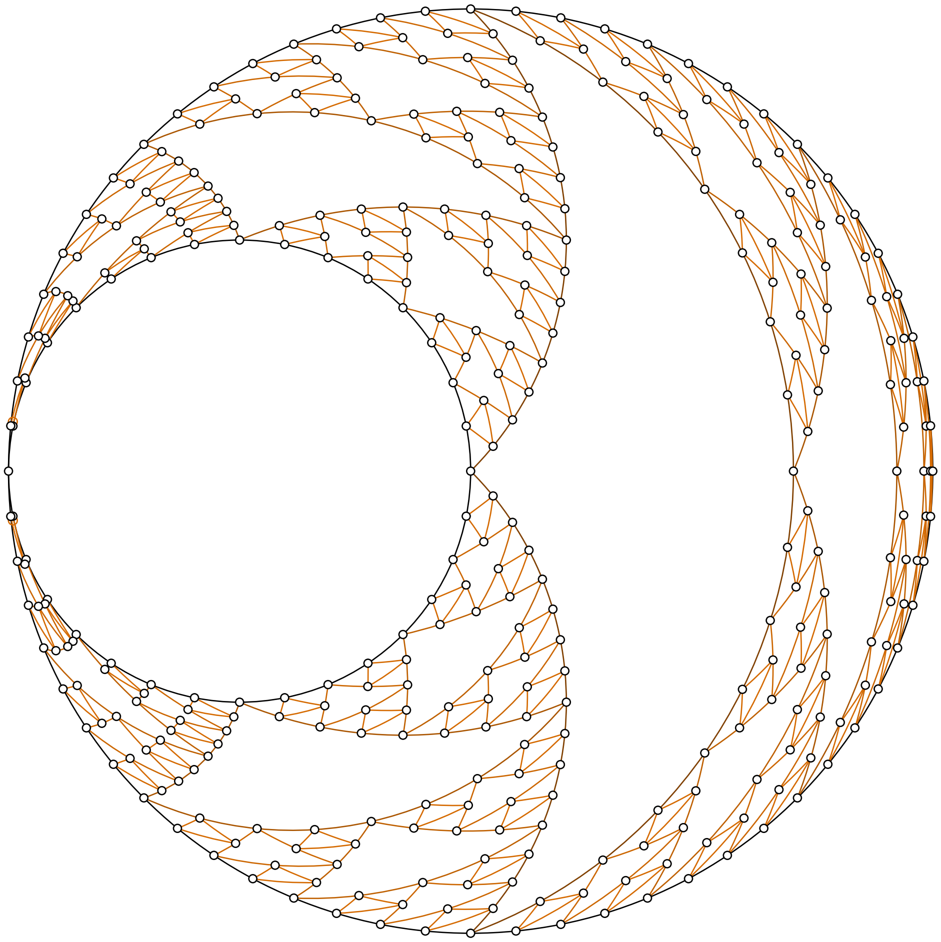
Beluhov
benchmark graphs
Jelliss
Lefebvre
Stappers
Wermuth
Knuth
KNUTH
Ewing

GASARCH

CONTENTS

I always thought Volume 4 was a myth,
like the missing part of the Dead Sea scrolls.

— BILL GASARCH (blog post, 10 January 2008) pinched gasket



July 1, 2025

A long train of consistent calculations opens itself out,
for every result of which there is found a corresponding
geometrical interpretation, in the theory of two of the celebrated
solids of antiquity, alluded to with interest by Plato in the Timæus;
namely, the Icosaedron, and the Dodecaedron.

— WILLIAM ROWAN HAMILTON (1856)

The total number of possible [knight's] tours that can be made is so vast that
it is safe to predict that no mathematician will ever
succeed in counting up the total.

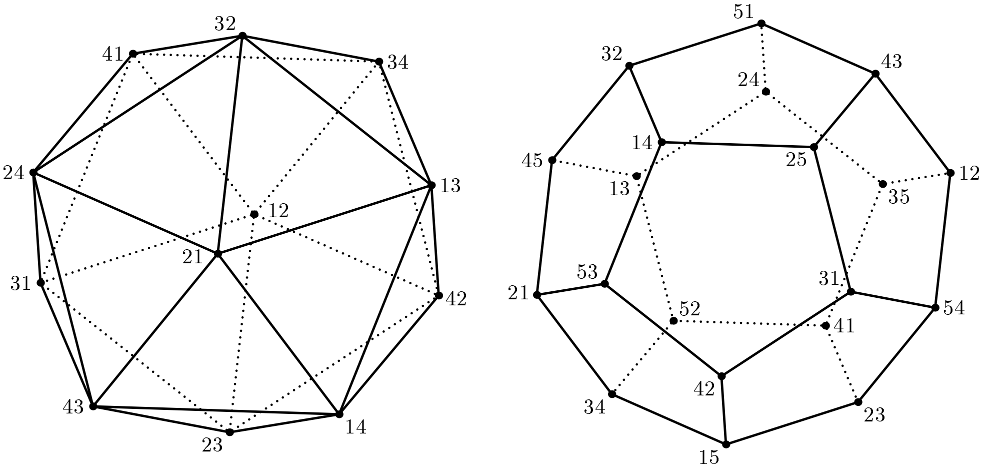
— ERNEST BERGHOLT (1915)

I'll show that this problem is susceptible to a very special analysis,
which merits extra attention because it involves reasoning of a kind rarely used
elsewhere. The excellence of Analysis is easy to see, but most people think that
it's limited to traditional questions about Mathematics; hence it will always be
quite important to apply Analysis to subjects that seem to make it out of reach,
for it incorporates the art of reasoning in the highest degree.
One cannot then extend the bounds of Analysis
without justifiably expecting great advantages.

— LEONHARD EULER (1759)

Plato
HAMILTON
BERGHOLT
EULER
Hamilton
quaternions
Platonic solids
icosahedron
Icosian Game
planar graph
dual of a planar graph
dodecahedron

7.2.2.4. Hamiltonian paths and cycles. A path or cycle that touches every vertex of a graph is called "Hamiltonian" in honor of W. R. Hamilton, who began to ponder and publicize such questions shortly after discovering the quaternions. Hamilton was fascinated by Platonic solids such as the icosahedron, with its 20 triangular faces; and he introduced what he called the Icosian Game, based on paths that go from face to face in that solid. Equivalently (see Fig. 121), his game was based on paths from vertices to vertices along the edges of a dodecahedron.



(a) Icosahedron; (b) Dodecahedron.

Fig. 121. The icosahedron and dodecahedron, whose vertices, edges, and faces define "dual" planar graphs: The faces of one solid correspond to the vertices of the other. (The vertices have been named with two-digit codes that are discussed in exercise 3.)

It's convenient to redraw Fig. 121(b) as three concentric rings, without crossing edges, as shown in Fig. 122(a). Then it's easy to find a Hamiltonian cycle, such as the one indicated by bold edges in Fig. 122(b). (In fact, Hamilton proved that *every* such cycle on the dodecahedron is essentially the same as this one; see exercise 9.) Thus we can also redraw the dodecahedron's graph as shown in Fig. 122(c). From that diagram it's *obviously* Hamiltonian — that is, it obviously has a spanning cycle; but it's not obviously planar at first glance.

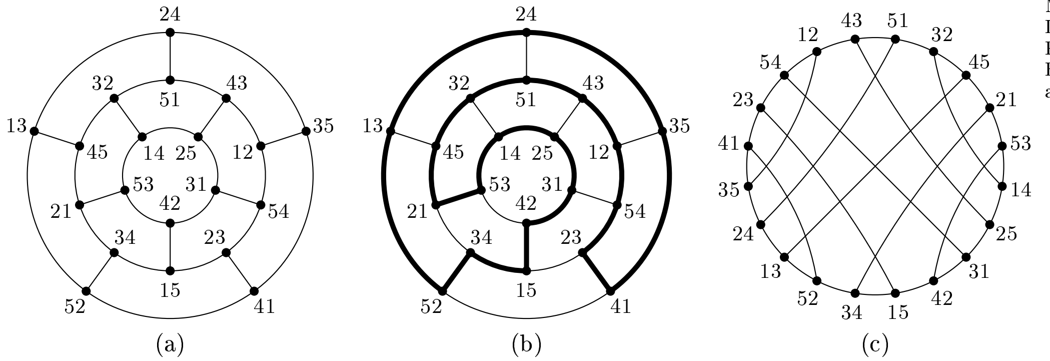


Fig. 122. Alternative views of a dodecahedron's vertices and edges.

Every Hamiltonian graph can clearly be drawn as a great big cycle, together with "chords" between certain pairs of vertices that aren't neighbors in the cycle. Thus a 3-regular graph can be specified compactly by listing only a third of its edges, if it is Hamiltonian. (On the other hand, many trivalent graphs are *not* Hamiltonian. In fact, the task of deciding whether or not a given cubic graph is Hamiltonian turns out to be NP-complete; see exercise 14.)

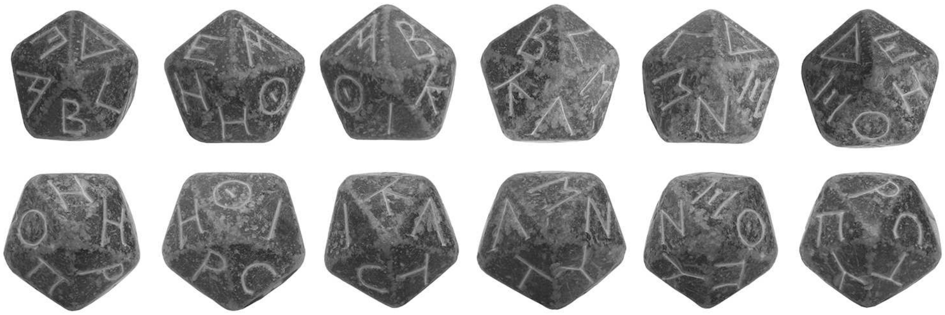
Hamiltonian paths in antiquity. Let's take a moment to discuss the rich history of the subject before we consider techniques by which Hamiltonian paths and cycles can be found. A strong case can actually be made for the assertion that questions of this kind represent the birth of graph theory, in the sense that they were the first nontrivial graph problems to be investigated.

For example, museums in many parts of the world contain specimens of ancient icosahedral objects whose 20 faces are inscribed with the first twenty letters of the Greek alphabet. In most of these cases the alphabetical sequence Α, Β, Γ, Δ, ..., Τ, Υ on such artifacts forms a Hamiltonian path between adjacent triangles. [E. Michon, in *Bulletin de la Société nationale des Antiquaires de France* (1897), 310 and (1904), 327–329, described an example in the Louvre, catalog number I 1532; P. Perdrizet, in *Bulletin de l'Institut français d'archéologie orientale* **30** (1930), 1–16, illustrated several others.]

In 2015, curators of the Egyptian antiquities at the British Museum kindly allowed the author to inspect the four icosahedra in their collection (EA 29418, EA 49738, EA 59731, EA 59732), of which the first three are Hamiltonian. The experience of rotating them by hand, slowly and systematically according to

concentric rings
Hamilton
spanning cycle
chords
3-regular graph
trivalent graphs
cubic graph
NP
Ham paths, history of–
Graeco-Roman icosahedra
Greek alphabet
Michon
Louvre
Perdrizet
British Museum
author

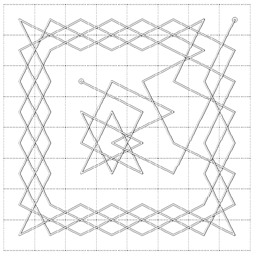
alphabetic order, turned out to be unexpectedly delightful. Here are views of the largest one, EA 49738, centered at each of its twelve vertices:



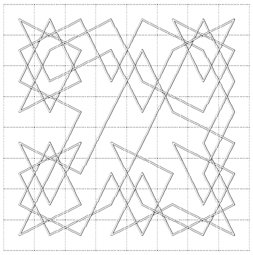
It is made of steatite, 5.8 centimeters in diameter and 228 grams in weight, and was acquired in 1911. A similar example, smaller and with more beautiful letterforms, is object number UC 59254 in the nearby Petrie Museum of University College London [see W. M. F. Petrie, *Objects of Daily Use* (1927), #288]. What a pleasant coincidence that W. R. Hamilton himself would independently come up with the same concept some 1800 years later, and would proceed to find a closed cycle instead of just a path!

Now fast forward to the ninth century, when Hamiltonian paths and cycles of quite a different kind came into play. The game of Chaturanga or Shaṭranj — a predecessor of chess, having different rules for certain pieces, but with knights moving just as they do today — was becoming popular in Asia. And in A.D. 842 the current world champion, al-‘Adlī ar-Rūmī, published a book about Shaṭranj. Complete copies of that work are lost; but we know from a subsequent treatise by Abū Zakarya Yaḥyā ibn Ibrāhīm al-Ḥakīm that al-‘Adlī had presented a closed ***knight's tour***: a Hamiltonian cycle on the chessboard. That same treatise also recorded an "open" knight's tour (a Hamiltonian path that can't be completed to a cycle), which was credited to an otherwise unknown author Ibn Manī‘.

Ibn Manī‘:



al-‘Adlī:



(1)

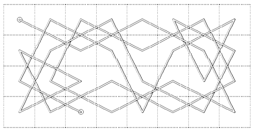
[See H. J. R. Murray, *A History of Chess* (Oxford, 1913), 175–176, 336.] These remarkable constructions are the earliest known solutions to what was destined to become a classic combinatorial problem. It seems likely that the first path was discovered before the first cycle, because there are so many more of the former.

alphabetic order
Pi, as written in Greece
University College London
Petrie
coincidence
Hamilton
reentrant knight's tour, see closed
Chaturanga
Shaṭranj
knights
al-‘Adlī ar-Rūmī
Abū Zakarya Yaḥyā
knight's tour
Ibn Manī‘
open versus closed tour
closed versus open tour
Murray

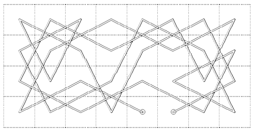
Rudraṭa
Ratnākara
slokas
fractured English

Remarkably, knight's tours on *half* of a chessboard, 4×8, had been published even earlier, by Kashmiri poets who were famous for their wordsmithing skills:

Rudraṭa
(c. 815)



Ratnākara
(c. 830)



(2)

Two copies of Rudraṭa's half-tour will make an open tour on the full board. And two copies of Ratnākara's will make a closed tour, if we rotate one copy by 180°.

Sanskrit poems traditionally consisted of verses called *slokas*, containing 32 syllables each. Here is sloka number 15 in chapter 5 of Rudraṭa's *Kāvyālaṅkāra*:

सेना लीलीलीना नाली लीनाना नानालीलीली । *senā līlīlīnā nālī līnānā nānālīlīlī*
नालीनालीले नालीना लीलीली नानानानाली ॥१५॥ *nālīnālīle nālīnā līlīlī nānānānālī* [15]

This enigmatic text, which speaks of military leadership, sounds almost like gibberish. But it cleverly represents a knight's tour, in the same way that his sloka 14 had represented a rook's tour: *When we read those 32 syllables in order of the left tour in* (2), *we get exactly the same words!*

More precisely, consider the following two 4×8 arrays of syllables σ_j:

$$\begin{array}{cccccccccccccccccc}
\sigma_1 & \sigma_{30} & \sigma_9 & \sigma_{20} & \sigma_3 & \sigma_{24} & \sigma_{11} & \sigma_{26} & \qquad & \sigma_1 & \sigma_2 & \sigma_3 & \sigma_4 & \sigma_5 & \sigma_6 & \sigma_7 & \sigma_8 \\
\sigma_{16} & \sigma_{19} & \sigma_2 & \sigma_{29} & \sigma_{10} & \sigma_{27} & \sigma_4 & \sigma_{23} & & \sigma_9 & \sigma_{10} & \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} & \sigma_{15} & \sigma_{16} \\
\sigma_{31} & \sigma_8 & \sigma_{17} & \sigma_{14} & \sigma_{21} & \sigma_6 & \sigma_{25} & \sigma_{12} & & \sigma_{17} & \sigma_{18} & \sigma_{19} & \sigma_{20} & \sigma_{21} & \sigma_{22} & \sigma_{23} & \sigma_{24} \\
\sigma_{18} & \sigma_{15} & \sigma_{32} & \sigma_7 & \sigma_{28} & \sigma_{13} & \sigma_{22} & \sigma_5 & & \sigma_{25} & \sigma_{26} & \sigma_{27} & \sigma_{28} & \sigma_{29} & \sigma_{30} & \sigma_{31} & \sigma_{32}
\end{array} \qquad (3)$$

The subscripts on the left correspond to the first sequence of knight moves in (2), while the subscripts on the right have their natural order. Rudraṭa composed a verse with the amazing property that both arrays agree (with $\sigma_1 = \sigma_1$, $\sigma_{30} = \sigma_2$, $\sigma_9 = \sigma_3$, $\ldots$, $\sigma_5 = \sigma_{32}$), by choosing $\sigma_1 =$ से, $\sigma_2 =$ ना, $\sigma_3 = \sigma_4 = \sigma_5 =$ ली, etc.

Notice that the constraints forced him to use at most four different symbols, thereby throwing away most of the tour's structure. It turns out, in fact, that there are *two* knight's tours consistent with his sloka. Therefore nobody knows whether he was thinking of the tour in (2) and (3) or the tour in exercise 36(i).

Thousands of 4×8 knight's tours are possible, and if Rudraṭa had known more of them he could have written a much less ambiguous sloka that had twelve distinct syllables. For example, a "fractured English" verse that describes such a tour might go like this (see exercise 36(ii)):

Want a good, good time, lots of fun?
Now not time so good; now not time.
Foo. Ah, so! So now fun is lost.
Time not now good, so time not now. (4)

Ratnākara came up with a better idea a few years later. For his tour, illustrated at the right of (2), he composed two *different* slokas, both of which made sense as part of his overall poem. Their syllable patterns

$$\begin{array}{cccccccccccccccccc}
\sigma_{26} & \sigma_{11} & \sigma_{24} & \sigma_5 & \sigma_{20} & \sigma_9 & \sigma_{30} & \sigma_7 & \qquad & \sigma_1 & \sigma_2 & \sigma_3 & \sigma_4 & \sigma_5 & \sigma_6 & \sigma_7 & \sigma_8 \\
\sigma_{23} & \sigma_4 & \sigma_{27} & \sigma_{10} & \sigma_{29} & \sigma_6 & \sigma_{19} & \sigma_{16} & & \sigma_9 & \sigma_{10} & \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} & \sigma_{15} & \sigma_{16} \\
\sigma_{12} & \sigma_{25} & \sigma_2 & \sigma_{21} & \sigma_{14} & \sigma_{17} & \sigma_8 & \sigma_{31} & & \sigma_{17} & \sigma_{18} & \sigma_{19} & \sigma_{20} & \sigma_{21} & \sigma_{22} & \sigma_{23} & \sigma_{24} \\
\sigma_3 & \sigma_{22} & \sigma_{13} & \sigma_{28} & \sigma_1 & \sigma_{32} & \sigma_{15} & \sigma_{18} & & \sigma_{25} & \sigma_{26} & \sigma_{27} & \sigma_{28} & \sigma_{29} & \sigma_{30} & \sigma_{31} & \sigma_{32}
\end{array} \qquad (5)$$

poetic license
Rudraṭa
Ratnākara
Bhoja
Deśika
Someshvara III
nonsense verse
doggerel
Yaḥyā ibn Ibrāhīm
abjad numerals

would have allowed him to define a tour quite precisely using 32 *distinct* syllables. (See his *Haravijaya*, Chapter 43, slokas 145 and 146.) For example, here's an English rendition of his two-sloka scheme:

Have some fun, watch this or that word—
Great four lines, take out, each gives eight.
Left; then two black; and just here white.
Three rook steps make one knight move, right?
(6)
One, two, three, four! Watch each word here;
Or take some left steps and move eight.
Just right gives this black rook great fun,
Then have lines make out that white knight.

We obtain the second verse by reading the first verse in knight's tour order, starting at the fifth syllable of the fourth line. (Ratnākara actually used only 24 different syllables. Furthermore, his choices for σ_5 and σ_{32} did not agree in the two slokas; this may be due to errors in transmission of the ancient text, or to "poetic license." In any case his remarkable poem clearly defined a knight's tour.)

Such wordplay had many devotees in medieval India. For example, Rudraṭa's tour of (2) was rendered in Ratnākara's two-sloka style by King Bhoja in his *Sarasvatī-kaṇṭhābharaṇa* (c. 1050), slokas 2.306 and 2.308; also by Vedānta Deśika in his devotional hymn *Pādukāsahasra* (1313), slokas 929 and 930.

A simpler scheme, capable of encoding knight's tours on the full 8×8 board, was used in slokas 5.623–632 of the encyclopedic Sanskrit work *Mānasollāsa* by King Someshvara III (c. 1130). He named each square of the board systematically by combining a consonant for the column with a vowel for the row; then an arbitrary tour was a nonsense verse of 64 syllables, which could be memorized if you wanted to impress your friends. For example, in English we could use the names

$$\begin{bmatrix} \text{bah} & \text{bay} & \text{bee} & \text{boe} & \text{boo} & \text{buh} & \text{bai} & \text{bao} \\ \text{dah} & \text{day} & \text{dee} & \text{doe} & \text{doo} & \text{duh} & \text{dai} & \text{dao} \\ \text{fah} & \text{fay} & \text{fee} & \text{foe} & \text{foo} & \text{fuh} & \text{fai} & \text{fao} \\ \text{hah} & \text{hay} & \text{hee} & \text{hoe} & \text{hoo} & \text{huh} & \text{hai} & \text{hao} \\ \text{lah} & \text{lay} & \text{lee} & \text{loe} & \text{loo} & \text{luh} & \text{lai} & \text{lao} \\ \text{mah} & \text{may} & \text{mee} & \text{moe} & \text{moo} & \text{muh} & \text{mai} & \text{mao} \\ \text{nah} & \text{nay} & \text{nee} & \text{noe} & \text{noo} & \text{nuh} & \text{nai} & \text{nao} \\ \text{sah} & \text{say} & \text{see} & \text{soe} & \text{soo} & \text{suh} & \text{sai} & \text{sao} \end{bmatrix} \qquad (7)$$

to encode Someshvara's tour as the following (memorable?) quatrain:

Sah nee soo nai lao fai bao duh, foe boo dee bah fay lah nay soe;
nuh sao mai hao dai huh doo bee, dah hay mah say noe suh nao lai.
Fao bai fuh dao buh doe bay fah, lay nah see noo sai mao hai foo?
Luh moe hee loo mee hoe moo lee, hoo fee boe day hah may loe muh. (8)

Incidentally, Yaḥyā ibn Ibrāhīm had presented the two knight's tours in (1) by first stating two 64-word poems in Arabic, then copying the words of those poems into 8×8 diagrams, according to the knight's paths. Then he repeated the right-hand tour, using the Arabic words "first," "second," ..., together with Persian-style abjad numerals, in place of the words of the corresponding

John Rylands Library
Path diagrams
dalla Volpe
greedy
heuristic
greedy algorithms
Warnsdorf–

poem. [His work is preserved in a rare manuscript belonging to the John Rylands Library in Manchester: *Arabic MS. 766*, folio 39.] The latter convention, which corresponds to

60	11	56	7	54	3	42	1
57	8	59	62	31	64	53	4
12	61	10	55	6	41	2	43
9	58	13	32	63	30	5	52
34	17	36	23	40	27	44	29
37	14	33	20	47	22	51	26
18	35	16	39	24	49	28	45
15	38	19	48	21	46	25	50

(9)

in decimal notation, has been used by many subsequent authors to characterize particular knight's tours in an easy-to-understand way. Path diagrams such as (1) and (2), which provide complementary insights, weren't invented until much later, when Lelio dalla Volpe published a short book *Corsa del Cavallo per tutt' i scacchi dello Scacchiere* (Bologna, 1766), containing nineteen examples.

A greedy heuristic. Early in the 1800s, the knight's tour problem inspired an important new approach to combinatorial problems, based on making a sequence of locally optimum decisions. Such techniques, now known as "greedy algorithms," were unheard-of at the time. But H. C. von Warnsdorf, a high court official in Hesse who had challenged himself by spending many nights trying to construct long paths of a knight, hit on a simple idea that worked like magic: *At each step, move to a place that has the fewest remaining exits.* This principle has become famous as "Warnsdorf's rule."

For example, suppose we want to construct an open knight's tour on a 5×5 board, starting in a corner. Numbering the cells ij for $0 \le i, j < 5$, we can assume by symmetry that the first two steps are 00 — 12. From cell 12 we can move the knight to either 04, 24, 33, 31, or 20, from which it could then exit in either 1, 3, 3, 3, or 3 ways; Warnsdorf's rule tells us to choose 04, because $1 < 3$. (Indeed, this is our last chance to visit 04, unless the tour will end at that cell.) After 04 the knight must proceed to 23; and again we have five choices, namely 44, 42, 31, 11, or 02. The rule takes us to 44, then 32; then to 40, then 21; and we've completed a partial tour that looks like this:

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

(**Bold** numbers are the visited cells. *Italicized* numbers tell how many exits remain from the unvisited cells.) (10)

Four cells are now candidates for step **9**, and they're all currently marked '*2*'. So there's a four-way tie. In such cases, von Warnsdorf explicitly said that it's OK to choose arbitrarily, among all cells that have the fewest exits. Let us therefore proceed boldly to cell 33 (between **4**, **5**, and **6**). That makes a two-way tie; and we might as well go next to 41 (just to the right of **7**). From here we *don't* want to go to the middle square, which has just dropped from *8* to *7*, because our

SGB format
linear time
greed
seven deadly sins
morality
virtue

other choice is a *1*. And now it's plain sailing, as von Warnsdorf leads us on a merry chase — ending gloriously with move **25** in the center cell 22.

It's easy to implement Warnsdorf's rule, by representing the given graph in SGB format. (The reader should be familiar with this format; see, for example, Algorithm 7B and the remarks that precede it.) The node for each vertex v in Algorithm W below extends the basic format by including two utility fields, $\mathtt{DEG}(v)$ and $\mathtt{TAG}(v)$, which correspond to the *italic* and **bold** numbers in (10).

Algorithm W allows the user to specify "target" vertices $t_1, \ldots, t_r$, which are to be visited only when no other vertices are available. A similar mechanism was, in fact, used by von Warnsdorf himself, in the advanced examples of his original booklet that introduced the idea [*Des Rösselsprunges einfachste und allgemeinste Lösung* (Schmalkalden, 1823); see also *Schachzeitung* **13** (1858), 489–492].

Algorithm W (*Warnsdorf's rule*). Given a graph G, a source vertex s, and optional target vertices $t_1, \ldots, t_r$, this algorithm applies Warnsdorf's rule to find a (hopefully Hamiltonian) path $v_1, v_2, \ldots$ that begins with s. Let $n = \mathtt{N}(G)$ be the number of vertices of G; let $v_0 = \mathtt{VERTICES}(G)$ be G's initial vertex in memory.

W1. [Initialize.] For $0 \le k < n$ and $v \leftarrow v_0 + k$, do the following: Set $d \leftarrow 0$, $a \leftarrow \mathtt{ARCS}(v)$; while $a \ne \Lambda$, set $d \leftarrow d + 1$ and $a \leftarrow \mathtt{NEXT}(a)$; then set $\mathtt{DEG}(v) \leftarrow d$ and $\mathtt{TAG}(v) \leftarrow 0$. (Thus $\mathtt{DEG}(v)$ is the degree of v.) Finally set $k \leftarrow 0$, $v \leftarrow s$, and $\mathtt{DEG}(t_i) \leftarrow \mathtt{DEG}(t_i) + n$ for $1 \le i \le r$.

W2. [Visit v.] Set $k \leftarrow k + 1$, $v_k \leftarrow v$, $\mathtt{TAG}(v) \leftarrow k$, $a \leftarrow \mathtt{ARCS}(v)$, and $\theta \leftarrow 2n$.

W3. [All arcs tested?] If $a = \Lambda$, go to W7. Otherwise set $u \leftarrow \mathtt{TIP}(a)$, and go to W6 if $\mathtt{TAG}(u) \ne 0$. (Vertex u is a neighbor of v_k and a candidate for v_{k+1}.)

W4. [Decrease $\mathtt{DEG}(u)$.] Set $t \leftarrow \mathtt{DEG}(u) - 1$ and $\mathtt{DEG}(u) \leftarrow t$.

W5. [Is $\mathtt{DEG}(u)$ smallest?] If $t < \theta$, set $\theta \leftarrow t$ and $v \leftarrow u$.

W6. [Loop over arcs.] Set $a \leftarrow \mathtt{NEXT}(a)$ and return to W3.

W7. [Done?] If $\theta = 2n$, terminate with path $v_1 \ldots v_k$. Otherwise go to W2. ▌

Notice that the candidates for v_{k+1} are precisely the vertices u whose DEG needs to change when v_k leaves the active graph. Therefore this algorithm runs in linear time: Every arc is examined at most twice, once in step W1 and once in step W3.

The path chosen by Algorithm W depends on the ordering of arcs that lead out of each vertex in SGB format, because Warnsdorf's rule makes an arbitrary decision in case of ties. A simple change to step W5 will randomize the path properly, as if all orderings of the arcs were equally likely (see exercise 53).

Now that we understand Warnsdorf's rule, let's talk a little bit about greed. Greed is of course one of the seven deadly sins; hence we might well question the morality of ever using a greedy algorithm in our own work. However, greed is actually a *virtue*, when it enhances the environment and harms nobody.

In what sense is Algorithm W greedy? From the standpoint of *short-term* greed, also known as "instant satisfaction," the best choice for v_{k+1} would seem to be a vertex with *maximum* degree, not minimum, because that vertex will give us the most flexibility when choosing v_{k+2}. But from the standpoint of *long-term* greed, also known as "risk management" or maximizing our chance of success, it's best to choose a vertex with *minimum* degree, as von Warnsdorf stipulated; that choice leaves us with the most arcs remaining for moves in the future. Indeed, short-term greed turns out to be very bad (see exercise 59).

How good is Warnsdorf's rule? It works so well for knight moves that von Warnsdorf naïvely believed it to be infallible, except perhaps on $m \times n$ boards with $m < 6$ or $n < 6$. He even thought that he had a proof of guaranteed success. His booklet exhibited many examples: $6\times 6, 6\times 7, \ldots,$ up to 10×10. Experiments by C. F. de Jaenisch [*Traité des applications de l'analyse mathématique au jeu des échecs* **2** (1862), 59] showed in fact that, on an ordinary 8×8 chessboard, one can basically choose the first 40 moves at random, and obtain a complete knight's tour by applying Warnsdorf's rule only to the last 24 steps!

The rule can fail, however. On a 6×6 board, it gives a complete tour about 97.2% of the time, yet it sometimes stops after only 32 or 34 steps if the starting position is one of the eight interior diagonal squares. On the 8×8 board it succeeds even more often (about 97.9%). Yet with probability 0.0000038 it might stop with a path of length 39, as shown in the answer to exercise 59.

Hamilton's dodecahedron graph (Fig. 122) is quite different from a graph of knight moves, because it is 3-regular. A partial path in a 3-regular graph can be extended in at most two ways, after we've selected the first two points, while a knight can have up to seven choices at every step. (Furthermore, all starting edges of the dodecahedron are equivalent.) Nevertheless, Algorithm W handles that graph well: It finds a Hamiltonian path $v_1v_2\ldots v_{20}$ with probability $\frac{31}{32} = .96875$. Furthermore, it finds a path with $v_{20} \;\frac{\quad}{}\; v_1$ (hence a Hamiltonian cycle) with probability $\frac{15}{128} \approx .117$. That probability rises to $\frac{139}{256} \approx .543$ if we set t_1 to a neighbor of s; it's exactly 1/2 if we set $\{t_1, t_2, t_3\}$ to the *three* neighbors of s.

It's not difficult to see that Algorithm W always works perfectly when G is the graph of a rectangular grid and s is a corner vertex (see exercise 62). With a bit more thought, we can even prove that it always succeeds when G is an n-cube, thereby finding many examples of the generalized Gray binary codes that we studied in Section 7.2.1.1 (see exercise 63). When G is the SGB graph *perms*$(-4, 0, 0, 0, 0, 0, 0)$ — whose vertices are the permutations of $\{0, 1, 2, 3, 4\}$, related by swapping adjacent digits — Warnsdorf's rule finds "change ringing" paths of length $5! - 1 = 119$ about 29% of the time. (See Algorithm 7.2.1.2P. This probability drops to less than 2%, however, with permutations of 6 elements, and to near zero with permutations of 7.) Another instructive example is the SGB graph *binary*$(10, 0, 0)$, whose vertices are the 16796 binary trees with 10 nodes, related by "rotation." Starting at the tree with all-null left links, Algorithm W finds a Hamiltonian path about 5.6% of the time. (See Algorithm 7.2.1.6L.)

Of course Algorithm W isn't a panacea. We can't expect any algorithm to solve the NP-complete Hamiltonian path problem in linear time! Warnsdorf's rule certainly has difficulty in critical cases; indeed, it can fail spectacularly even on small graphs (see exercise 65). But it's often a good first thing to try, when presented with a graph that we haven't seen before.

Ira Pohl [*CACM* **10** (1967), 446–449; **11** (1968), 1] has suggested breaking ties in Warnsdorf's rule by looking at the *sum of the degrees* of v_k's neighbors.

de Jaenisch
Hamilton
dodecahedron graph
3-regular
n-cube
Gray binary codes
perms
change ringing
binary
binary trees
rotation
NP-complete
Pohl

Path flipping. Long before Warnsdorf's time, the great mathematician Leonhard Euler had already published a classic paper about knight's tours [*Mémoires de l'académie des sciences de Berlin* **15** (1759), 310–337], in which he showed how to discover long paths by a completely different method. (Euler credited this idea, at least in part, to his friend Louis Bertrand.) Instead of Warnsdorf's "greedy" algorithm, his approach might be called a "breedy" method, because it proceeded by simple mutations and adaptations of paths already known.

path flipping–
flipping paths–
path exchange, see path flipping
Euler
Bertrand
breedy
mutations
genetic algorithms
closed

Suppose, for example, that we want to find a 3×10 knight's tour, and that Warnsdorf has already told us how to reach 28 of the 30 cells:

4	7	2	27	24	13	10	19	a	17
1	28	5	14	9	22	25	16	11	20
6	3	8	23	26	15	12	21	18	b

. (11)

We can't go from position 28 to an unvisited cell; but we needn't despair, because 28 is just one knight's move away from cell 23. Similarly, cell 1 is adjacent to 8. Therefore we can immediately deduce that two more equally long paths exist:

$$1\,..\,23, 28\,..\,24; \qquad 7\,..\,1, 8\,..\,28. \tag{12}$$

(Here '$x\,..\,y$' stands for the path from x to y that proceeds by unit steps ± 1.) Operating in the same fashion on the first of these yields three more,

$$1\,..\,5, 24\,..\,28, 23\,..\,6; \;\; 1\,..\,15, 24\,..\,28, 23\,..\,16; \;\; 7\,..\,1, 8\,..\,23, 28\,..\,24. \tag{13}$$

And, aha, one of these can be extended to a full tour $1\,..\,15, 24\,..\,28, 23\,..\,16, \mathrm{b}, \mathrm{a}$:

4	7	2	19	16	13	10	25	30	27
1	20	5	14	9	22	17	28	11	24
6	3	8	21	18	15	12	23	26	29

. (14)

Now the same subpath-flipping technique leads from (14) to additional tours

$$1\,..\,17, 30\,..\,18; \qquad 1\,..\,23, 30\,..\,24; \qquad 7\,..\,1, 8\,..\,30; \tag{15}$$

and we can continue to find tours galore:

$$1\,..\,13, 18\,..\,30, 17\,..\,14; \quad 1\,..\,5, 18\,..\,30, 17\,..\,6; \quad 7\,..\,1, 8\,..\,17, 30\,..\,18;$$
$$7\,..\,1, 8\,..\,23, 30\,..\,24; \quad 13\,..\,8, 1\,..\,7, 14\,..\,30; \quad 1\,..\,7, 14\,..\,17, 30\,..\,18, 13\,..\,8;$$

etc. Indeed, the latter is a Hamiltonian *cycle* — a *closed* tour — because 1 is adjacent to 8! A Hamiltonian cycle represents 30 different Hamiltonian *paths*, each of which leads to further flips, hence further paths and cycles.

If we start with (14) and keep flipping until no new paths arise, it turns out that we will have discovered all 16 of the Hamiltonian cycles of the 3×10 knight graph, as well as 2472 of its 2568 noncyclic Hamiltonian paths.

One of the 96 noncyclic Hamiltonian paths *not* derivable from (14) is

1	12	3	22	15	10	7	26	29	18
4	23	14	11	6	25	20	17	8	27
13	2	5	24	21	16	9	28	19	30

. (16)

It leads via flips only to three others, namely to $1\,..\,17, 30\,..\,18$; $13\,..\,1, 14\,..\,30$; $13\,..\,1, 14\,..\,17, 30\,..\,18$. We *wouldn't* have found a cycle, if we'd started with (16).

Euler
breadth-first search
update
canonical form
breadth-first search

Let's formulate Euler's approach more precisely:

Algorithm F (*Long paths by flipping*). Given a simple path $v_1 \mathrel{-\!-} v_2 \mathrel{-\!-} \cdots \mathrel{-\!-} v_t$ in a connected n-vertex graph, this algorithm repeatedly obtains new paths by reversing subpaths as explained above, until either exhausting all possibilities or finding a path that can be extended by a vertex $\notin \{v_1, v_2, \ldots, v_t\}$. An auxiliary table of vertex labels $w[v]$ is used to discover potential flips.

F1. [Initialize for breadth-first search.] Prepare a dictionary, initially empty, for storing paths of length $t-1$. Set $w[v] \leftarrow 0$ for each of the n vertices v of the graph. Set $q \leftarrow 0$ and perform $\mathit{update}(v_1, \ldots, v_t)$, where *update* is the subroutine defined below. Then set $d \leftarrow p \leftarrow p_1 \leftarrow p_2 \leftarrow 0$ and $p_0 \leftarrow q$.

F2. [Done with distance d?] (At this point we've entered q paths into the dictionary, and we've explored the successors of the first p paths. Exactly p_i of those paths were obtained by making $\le d-i$ flips, for $0 \le i \le 2$.) Go to F6 if $p = p_0$; otherwise set $p \leftarrow p+1$.

F3. [Explore path p.] Let $u_1 \mathrel{-\!-} u_2 \mathrel{-\!-} \cdots \mathrel{-\!-} u_t$ be the pth path that entered the dictionary, and set $w[u_k] \leftarrow k$ for $1 \le k \le t$. Go to F5 if $u_t \mathrel{-\!-} u_1$.

F4. [Process a noncyclic path.] For each vertex v such that $u_t \mathrel{-\!-} v$, do the following: Set $k \leftarrow w[v]$; terminate the algorithm if $k = 0$; otherwise call $\mathit{update}(u_1, \ldots, u_k, u_t, \ldots, u_{k+1})$. Then, for each v such that $u_1 \mathrel{-\!-} v$, do the following: Set $k \leftarrow w[v]$; terminate if $k = 0$; otherwise $\mathit{update}(u_{k-1}, \ldots, u_1, u_k, \ldots, u_t)$. Then return to F2.

F5. [Process a cyclic path.] (A cyclic path will be in the dictionary only if $t = n$; see below.) For $1 \le j \le t$ and for each v such that $u_j \mathrel{-\!-} v$, do the following: Set $k \leftarrow w[v]$ (which will be positive). If $k < j$, call $\mathit{update}(u_{j+1}, \ldots, u_t, u_1, \ldots, u_k, u_j, \ldots, u_{k+1})$ and $\mathit{update}(u_{k-1}, \ldots, u_1, u_t, \ldots, u_j, u_k, \ldots, u_{j-1})$; otherwise $\mathit{update}(u_{j+1}, \ldots, u_k, u_j, \ldots, u_1, u_t, \ldots, u_{k+1})$ and $\mathit{update}(u_{k-1}, \ldots, u_j, u_k, \ldots, u_t, u_1, \ldots, u_{j-1})$. Then return to F2.

F6. [Advance d.] Terminate if $p = q$ (we have found all the reachable paths). Otherwise set $d \leftarrow d+1$, $p_2 \leftarrow p_1$, $p_1 \leftarrow p_0$, $p_0 \leftarrow q$, and go back to F2. ▌

Algorithm F relies on a subroutine '$\mathit{update}(v_1, \ldots, v_t)$', whose purpose is to put the path $v_1 \mathrel{-\!-} \cdots \mathrel{-\!-} v_t$ into the dictionary unless it's already there. First the path is converted to a canonical form, so that equivalent paths are entered only once: If $v_t \mathrel{-\!\!\!/\!\!\!-} v_1$, the canonical form is obtained by changing $(v_1, \ldots, v_t) \leftarrow (v_t, \ldots, v_1)$ if $v_1 > v_t$. On the other hand if $v_t \mathrel{-\!-} v_1$, the path is cyclic, and we terminate the algorithm if $t < n$. (The graph is connected, so there must be a vertex outside the cycle that is adjacent to a vertex of the cycle.) Finally, if $t = n$ and $v_n \mathrel{-\!-} v_1$, we obtain the canonical form by permuting the cycle cyclically so that v_1 is the smallest element; then we set $(v_1, v_2, \ldots, v_n) \leftarrow (v_1, v_n, \ldots, v_2)$ if $v_2 > v_n$. Once $(v_1, \ldots, v_t)$ is in canonical form, the *update* routine looks for it in the dictionary. If unsuccessful, *update* sets $q \leftarrow q+1$ and inserts it as the qth path.

The theory of breadth-first search tells us that $(v_1, \ldots, v_t)$ cannot match any path in the dictionary that was obtained with fewer than $d-1$ flips. (Otherwise the path $(u_1, \ldots, u_t)$ that led to it would have been seen before making d flips.)

Therefore step F6 can save dictionary space and lookup time by deleting all paths of index $\le p_2$ from the dictionary whenever p_2 increases. Exercise 73 discusses a simple trick that makes this deletion painless.

Algorithm F is amazingly versatile. For example, there are 9862 closed knight's tours on a 6×6 board, and 2963928 open tours. All of them will be found by Algorithm F, when given any single instance.

We began our search for a 3×10 knight's cycle by using the Warnsdorf-inspired path (11). But we could have started Algorithm F with $t = 1$, thus presenting it with only a single vertex v_1. Every time the algorithm finds a larger path, we can simply restart it, with t increased.

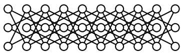
For example, the author tried the 3×10 problem 100 times, choosing v_1 at random and ordering the vertex neighbors randomly in steps F4 and F5. A Hamiltonian cycle was found in 82 cases, usually after making fewer than 100 calls on *update*. A stubborn Hamiltonian path like (16) was found in 6 cases. And the remaining 12 cases failed to reach $t = 30$; once t was even stuck at 22.

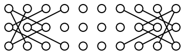
Of course that's a very small problem. When presented with the graph of permutations of $\{0, 1, 2, 3, 4, 5\}$, Algorithm F was able to find a "change ringing" cycle of length 720 in each of ten random trials, averaging less than 50,000 updates per trial. On the other hand it did *not* do well when trying to find a closed 3×100 knight's tour.

Searching exhaustively. Let's try now to design an algorithm that systematically finds *every* Hamiltonian cycle of a given graph. Such an algorithm will also find every Hamiltonian *path*, because exercise 2 shows that every Hamiltonian path of G corresponds to a Hamiltonian cycle of a related graph G'.

A Hamiltonian cycle involves every vertex. So we can start it at any convenient vertex v_1. Then there's an obvious way to grow all possible cycles via backtracking: For each v_2 with $v_1 \mathrel{-\!\!-} v_2$, we consider each $v_3 \ne v_1$ with $v_2 \mathrel{-\!\!-} v_3$, etc. We can also formulate the task as an XCC problem (see, for example, the "prime queen attacking problem" in Section 7.2.2.3).

But those approaches are overly specific; there's usually a much more efficient way to proceed. Instead of regarding our task as the assignment of appropriate labels 1, 2, …, n to the n vertices of our graph, it's better to regard it as the task of choosing n edges in such a way that (i) every vertex is an endpoint of precisely two of those edges; (ii) the subgraph defined by those edges is connected. Indeed, a Hamiltonian cycle is nothing more nor less than an (unordered) set of n edges that form a single cycle.

Consider, for example, the 3×10 knight graph, , which has 30 vertices and 50 edges. We can conveniently denote its vertices by two-digit numbers, $\{00, 01, \ldots, 09, 10, 11, \ldots, 19, 20, 21, \ldots, 29\}$, and write its edges compactly in two-line form: $\genfrac{}{}{0pt}{}{00}{12}, \genfrac{}{}{0pt}{}{00}{21}, \genfrac{}{}{0pt}{}{01}{20}, \ldots, \genfrac{}{}{0pt}{}{17}{25}, \genfrac{}{}{0pt}{}{17}{29}, \genfrac{}{}{0pt}{}{18}{26}, \genfrac{}{}{0pt}{}{19}{27}$.

Notice that vertices 00, 09, 10, 11, 18, 19, 20, and 29 have degree 2 in this graph. Consequently the two edges that touch each of them *must* be present in any Hamiltonian cycle; in other words, the pattern  is already forced, before we begin to choose further edges.

Warnsdorf
permutations
change ringing
all Hamiltonian cycles
all Hamiltonian paths
backtracking
XCC problem
prime queen attacking problem
knight graph

In this pattern, vertex 12 belongs to the edges ${}^{00}_{12}$ and ${}^{12}_{20}$, which were forced from vertices 00 and 20. So 12 already has the two edges that it needs; and the other edges that touch it, namely ${}^{04}_{12}$ and ${}^{12}_{24}$, cannot ever be part of a Hamiltonian cycle. We might as well delete them. Similarly, we can delete edges ${}^{05}_{17}$ and ${}^{17}_{25}$.

Nothing else is obviously forced at this point, so we must make a choice. For example, we must use either the edge ${}^{02}_{21}$ or the edge ${}^{13}_{21}$, because those edges are the only ways for vertex 21 to reach its quota of two. In other words, our search tree for the 3 × 10 knight graph must begin with a binary branch.

Suppose we choose ${}^{02}_{21}$. That edge implies in particular that we now have $01 \mathrel{-\!\!-} 20 \mathrel{-\!\!-} 12 \mathrel{-\!\!-} 00 \mathrel{-\!\!-} 21 \mathrel{-\!\!-} 02 \mathrel{-\!\!-} 10 \mathrel{-\!\!-} 22$ as a subpath of the final cycle. Consequently edges ${}^{13}_{21}$, ${}^{02}_{14}$, and ${}^{02}_{23}$ can no longer be used. Nor can the edge ${}^{01}_{22}$ (because it would complete a short cycle!).

Aha. Only one of the remaining edges touches 01, namely ${}^{01}_{13}$. So that edge is now forced. And then we're confronted with another two-way branch. Exercise 108 discusses one sequence of reasonable initial choices, and the reader is strongly encouraged to study that scenario.

In general, as we're trying to visit all Hamiltonian cycles of a given graph, we'll have a partial solution consisting of a set of disjoint subpaths to be included, and a set of edges by which those subpaths might be extended until a complete cycle is obtained. The subpaths are defined by the edges that have been chosen so far. If there are t subpaths, $\{u_1 \ldots v_1, \ldots, u_t \ldots v_t\}$, we say that the $2t$ endpoints $\{u_1, v_1, \ldots, u_t, v_t\}$ are "outer" vertices; any vertex that lies on a subpath but is *not* an endpoint is called "inner"; and all other vertices are "bare." Every vertex begins bare, and is eventually clothed. If we reach a state where all but two of the vertices are inner, and if those two outer vertices are adjacent, we can complete a Hamiltonian cycle. Success!

Algorithm H below finds Hamiltonian cycles by essentially starting with a graph G and removing edges until only a cycle remains. It uses a sparse-set representation for G, because such structures are an especially attractive way to maintain the current status of a graph that is continually getting smaller.

The idea is to have two arrays, NBR and ADJ, with one row for each vertex v. If v has $d = \texttt{DEG}(v)$ neighbors in G, they're listed (in any order) in the first d columns of NBR[v]. And if NBR[v][k] $= u$, where $0 \le k < d$, we have ADJ[v][u] $= k$; in other words, there's an important invariant relation,

$$\texttt{NBR}[v]\,[\texttt{ADJ}[v]\,[u]] = u, \qquad \text{for } 0 \le u < n, \tag{17}$$

where n is the number of vertices in G. Neighbors can be deleted by moving them to the right and decreasing d; neighbors can be undeleted by simply increasing d. Furthermore, if u is *not* a neighbor of v, ADJ[v][u] has the impossible value ∞; thus the ADJ array functions also as an adjacency matrix.

The edges $u \mathrel{-\!\!-} v$ of G are considered to be pairs of arcs, $u \longrightarrow v$ and $v \longrightarrow u$, which run in opposite directions. In particular, we always have ADJ[u][v] $= \infty$ if and only if ADJ[v][u] $= \infty$. When an edge is deleted, however, we often need to delete only one of those arcs in the NBR array, because Algorithm H doesn't always need to look at both of them.

binary branch: A choice between two possibilitie
outer vertices
inner vertices
bare vertices
sparse-set representation
data structures
invariant relation
adjacency matrix

Algorithm H represents the vertices of G by the integers 0 to $n-1$, as we've seen in (17). Every vertex v has three fields: its current degree, $\texttt{DEG}(v)$; its external name, $\texttt{NAME}(v)$, used only to print the answers; and a special field called $\texttt{MATE}(v)$. We have $\texttt{MATE}(u)=v$ and $\texttt{MATE}(v)=u$ when u and v are the outer vertices at the ends of a current subpath; and we have $\texttt{MATE}(v)=-1$ if and only if vertex v is bare. The value of $\texttt{MATE}(v)$ is undefined when v is an inner vertex, but it must be nonnegative in that case. (See exercise 109.)

An inner vertex is essentially invisible to our algorithm, because we already know its context in the final cycle. We maintain an array VIS to list the visible vertices — those that are either bare or outer. VIS is a sparse-set representation, containing a permutation of the vertices, with the invisible ones listed last. The inverse permutation appears in a companion array called IVIS, so that we have

$$\texttt{VIS[}k\texttt{]}=v \qquad\iff\qquad \texttt{IVIS[}v\texttt{]}=k. \tag{18}$$

Vertex v is visible if and only if $\texttt{IVIS[}v\texttt{]}<\texttt{S}$, where S is a global variable. Thus v is inner if and only if $\texttt{IVIS[}v\texttt{]}\ge\texttt{S}$. Here's how a vertex becomes invisible:

$$\text{make\_inner}(v)=\begin{cases}\text{Set } \texttt{S}\leftarrow\texttt{S}-1,\ v'\leftarrow\texttt{VIS[S]},\ k\leftarrow\texttt{IVIS[}v\texttt{]};\\ \text{set } \texttt{VIS[S]}\leftarrow v,\ \texttt{IVIS[}v\texttt{]}\leftarrow\texttt{S},\\ \qquad\texttt{VIS[}k\texttt{]}\leftarrow v',\ \texttt{IVIS[}v'\texttt{]}\leftarrow k.\end{cases} \tag{19}$$

If u is a bare vertex whose degree decreases to 2, Algorithm H can make significant progress, because the two remaining edges that touch u must both be part of the cycle. Whenever such a u is discovered, we put it into a "trigger list" called TRIG. A global variable, T, holds the size of the trigger list. This behavior is implemented by using the following procedure to delete the arc $u \longrightarrow v$:

$$\text{remarc}(u,v)=\begin{cases}\text{Set } d\leftarrow\texttt{DEG}(u)-1,\ k\leftarrow\texttt{ADJ[}u\texttt{][}v\texttt{]},\ w\leftarrow\texttt{NBR[}u\texttt{][}d\texttt{]}.\\ \text{If } \texttt{MATE}(u)<0 \text{ and } d=2, \text{ set } \texttt{TRIG[T]}\leftarrow u,\ \texttt{T}\leftarrow\texttt{T}+1.\\ \text{Set } \texttt{NBR[}u\texttt{][}d\texttt{]}\leftarrow v,\ \texttt{NBR[}u\texttt{][}k\texttt{]}\leftarrow w,\\ \qquad\texttt{ADJ[}u\texttt{][}v\texttt{]}\leftarrow d,\ \texttt{ADJ[}u\texttt{][}w\texttt{]}\leftarrow k;\\ \text{set } \texttt{DEG}(v)\leftarrow d.\end{cases} \tag{21}$$

One might think that we've now defined a comprehensive set of data structures for implementing Algorithm H; but we aren't done yet. There's also a doubly linked list, maintained in arrays $\texttt{LLINK[}v\texttt{]}$ and $\texttt{RLINK[}v\texttt{]}$ for $0\le v\le n$, with entries $\texttt{LLINK[}n\texttt{]}$ and $\texttt{RLINK[}n\texttt{]}$ serving as the list head. This list contains all of the current "outer" vertices. More precisely, suppose that there are t subpaths, and suppose that we have $\texttt{RLINK[}n\texttt{]}=v_1$, $\texttt{RLINK[}v_j\texttt{]}=v_{j+1}$ for $1\le j<2t$, and $\texttt{RLINK[}v_{2t}\texttt{]}=n$. Then the outer vertices are $\{v_1,v_2,\ldots,v_{2t}\}$; and we also have $\texttt{LLINK[}n\texttt{]}=v_{2t}$, $\texttt{LLINK[}v_j\texttt{]}=v_{j-1}$ for $1<j\le 2t$, and $\texttt{LLINK[}v_1\texttt{]}=n$. Insertion and deletion are accomplished in the usual way:

$$\text{activate}(v)=\begin{cases}\text{Set } k\leftarrow\texttt{LLINK[}n\texttt{]},\\ \quad\texttt{LLINK[}n\texttt{]}\leftarrow\texttt{RLINK[}k\texttt{]}\leftarrow v,\\ \quad\texttt{LLINK[}v\texttt{]}\leftarrow k,\ \texttt{RLINK[}v\texttt{]}\leftarrow n.\end{cases} \tag{22}$$

$$\text{deactivate}(v)=\begin{cases}\text{Set } j\leftarrow\texttt{LLINK[}v\texttt{]},\ k\leftarrow\texttt{RLINK[}v\texttt{]},\\ \quad\texttt{LLINK[}k\texttt{]}\leftarrow j,\ \texttt{RLINK[}j\texttt{]}\leftarrow k;\\ \text{make\_inner}(v).\end{cases} \tag{23}$$

degree
$\texttt{DEG}(v)$
$\texttt{NAME}(v)$
$\texttt{MATE}(v)$
outer vertices
bare
inner vertex
sparse-set representation
visible
invisible
make_inner(v)
trigger list
data structures
doubly linked list
$\texttt{LLINK[}v\texttt{]}$
$\texttt{RLINK[}v\texttt{]}$
list head
outer vertices
Insertion
deletion

The algorithm makes frequent use of the following subroutine:

stack with holes
n-cycle

$$\text{makemates}(u,w) = \begin{cases} \text{If } \texttt{ADJ[}w\texttt{][}u\texttt{]} < \texttt{DEG(}w\texttt{)},\\ \quad \text{remarc}(u,w) \text{ and remarc}(w,u);\\ \text{set } \texttt{MATE(}u\texttt{)} \leftarrow w \text{ and } \texttt{MATE(}w\texttt{)} \leftarrow u. \end{cases} \qquad (24)$$

Vertices u and w are becoming endpoints. It removes the edge between u and w, if present, in order to prevent the formation of a short (non-Hamiltonian) cycle.

The current state at each level of the search tree is kept in two sequential stacks. ACTIVE is an array that remembers which vertices are outer; SAVE is an array that remembers the mates and degrees of the visible vertices. The SAVE stack is somewhat unusual because it's a "stack with holes": n slots are allocated to it at every level, but only the slots for visible vertices are actually used.

Level l of the search can in fact involve up to seven state variables: $\texttt{CV(}l\texttt{)}$ is the outer vertex on which we're branching; $\texttt{I(}l\texttt{)}$ identifies the neighbor of that vertex in the currently chosen edge; $\texttt{D(}l\texttt{)}$ is the current degree of $\texttt{CV(}l\texttt{)}$; $\texttt{E(}l\texttt{)}$ is the number of edges chosen so far; $\texttt{S(}l\texttt{)}$ is the number of vertices that are currently visible; $\texttt{T(}l\texttt{)}$ and $\texttt{A(}l\texttt{)}$ are the current sizes of TRIG and ACTIVE.

Algorithm H (*All Hamiltonian cycles*). Given a graph G on the n vertices $\{0,1,\ldots,n-1\}$, this algorithm uses the data structures discussed above to visit every subset of n edges that form an n-cycle. During every visit, the chosen edges are $\texttt{EU[}k\texttt{]} \mathrel{—\!—} \texttt{EV[}k\texttt{]}$ for $0 \le k < n$.

H1. [Initialize.] Set up the NBR and ADJ arrays as described in (17). Also set $\texttt{VIS[}v\texttt{]} \leftarrow \texttt{IVIS[}v\texttt{]} \leftarrow v$ and $\texttt{MATE(}v\texttt{)} \leftarrow -1$ for $0 \le v < n$. Set the global variables $a \leftarrow e \leftarrow i \leftarrow l \leftarrow \texttt{T} \leftarrow 0$. Set $\texttt{LLINK[}n\texttt{]} \leftarrow \texttt{RLINK[}n\texttt{]} \leftarrow \texttt{S} \leftarrow n$. Finally, for every vertex v with $\texttt{DEG(}v\texttt{)} = 2$, set $\texttt{TRIG[T]} \leftarrow v$ and $\texttt{T} \leftarrow \texttt{T}+1$.

H2. [Choose the root vertex.] Let CURV be a vertex of minimum degree, and set $d \leftarrow \texttt{DEG(CURV)} - 1$. If $d < 1$, terminate (there is no Hamiltonian cycle). If $d = 1$, set $\texttt{CURV} \leftarrow -1$ and go to H4.

H3. [Force a root edge.] Set $\texttt{CURU} \leftarrow \texttt{NBR[CURV][}d-i\texttt{]}$ (the last yet-untried neighbor of CURV), and set $\texttt{EU[0]} \leftarrow \texttt{CURU}$, $\texttt{EV[0]} \leftarrow \texttt{CURV}$, $e \leftarrow 1$. Then activate(CURU), activate(CURV), and makemates(CURU, CURV).

H4. [Record the state.] Set $\texttt{CV(}l\texttt{)} \leftarrow \texttt{CURV}$, $\texttt{I(}l\texttt{)} \leftarrow i$, $\texttt{D(}l\texttt{)} \leftarrow d$, $\texttt{E(}l\texttt{)} \leftarrow e$, $\texttt{S(}l\texttt{)} \leftarrow \texttt{S}$, $\texttt{T(}l\texttt{)} \leftarrow \texttt{T}$. For $0 \le k < \texttt{S}$, set $u \leftarrow \texttt{VIS[}k\texttt{]}$ and $\texttt{SAVE[}nl+u\texttt{]} \leftarrow \bigl(\texttt{MATE(}u\texttt{)},\texttt{DEG(}u\texttt{)}\bigr)$ (thereby leaving "holes" in the SAVE stack). Then set $u \leftarrow \texttt{RLINK[}n\texttt{]}$; while $u \ne n$ set $\texttt{ACTIVE[}a\texttt{]} \leftarrow u$, $a \leftarrow a+1$, $u \leftarrow \texttt{RLINK[}u\texttt{]}$. Finally set $\texttt{A(}l\texttt{)} \leftarrow a$, and go to H6 if $l = 0$.

H5. [Choose an edge.] Set $\texttt{CURU} \leftarrow \texttt{NBR[CURV][}i\texttt{]}$, $\texttt{CURT} \leftarrow \texttt{MATE(CURU)}$, $\texttt{CURW} \leftarrow \texttt{MATE(CURV)}$, $\texttt{EU[}e\texttt{]} \leftarrow \texttt{CURU}$, and $e \leftarrow e+1$. If $\texttt{CURT} < 0$ (CURU is bare), makemates(CURU, CURW), activate(CURU), and go to H6. Otherwise (CURU is outer), makemates(CURT, CURW). Call $\text{remarc}\bigl(\texttt{NBR[CURU][}k\texttt{]},\texttt{CURU}\bigr)$ for k decreasing from $\texttt{DEG(CURU)} - 1$ to 0. Then deactivate(CURU).

H6. [Begin trigger loop.] Set $j \leftarrow 0$ if $l = 0$, else $j \leftarrow \texttt{T(}l{-}1\texttt{)}$. Go to H10 if $j = \texttt{T}$.

data structures

H7. [Clothe $\mathtt{TRIG}[j]$.] Set $v \gets \mathtt{TRIG}[j]$, and go to H9 if $\mathtt{MATE}(v) \ge 0$ (v is no longer bare). Otherwise go to H15 if $\mathtt{DEG}(v) < 2$ (Hamiltonian cycle is impossible). Set $u \gets \mathtt{NBR}[v]\,[0]$ and $v \gets \mathtt{NBR}[v]\,[1]$. Go to H15 if $w = \mathtt{MATE}(u)$ and $e \ne n-2$ (cycle is too short). Set $\mathtt{EU}[e] \gets u$, $\mathtt{EV}[e] \gets v$, $e \gets e+1$, $\mathtt{EU}[e] \gets v$, $\mathtt{EV}[e] \gets w$, $e \gets e+1$, $\mathtt{MATE}(v) \gets v$, and make_inner(v).

H8. [Take stock.] (We've just joined v to its only two neighbors, u and w, which aren't mates unless $e = n$.) Update the data structures as described in exercise 112, based on whether $\mathtt{MATE}(u){<}0$ and/or $\mathtt{MATE}(w){<}0$ (four cases).

H9. [End trigger loop?] Set $j \gets j+1$, and return to H7 if $j < \mathtt{T}$.

H10. [Enter new level.] Set $l \gets l+1$, and go to H13 if $e \ge n-1$.

H11. [Choose vertex for branching.] Set $\mathtt{CURV} \gets \mathtt{RLINK}[n]$, $d \gets \mathtt{DEG(CURV)}$, $k \gets \mathtt{RLINK[CURV]}$. While $k \ne n$, if $\mathtt{DEG}(k) < d$ reset $\mathtt{CURV} \gets k$ and $d \gets \mathtt{DEG}(k)$; set $k \gets \mathtt{RLINK}[k]$. Go to H14 if $d = 0$. Otherwise set $\mathtt{EV}[e] \gets \mathtt{CURV}$ and $\mathtt{T} \gets \mathtt{T}(l-1)$. (See exercise 129.)

H12. [Make $\mathtt{CURV}$ inner.] Call remarc$\bigl(\mathtt{NBR[CURV]}[k],\mathtt{CURV}\bigr)$ for $0 \le k < d$ (thereby removing $\mathtt{CURV}$ from its neighbors' lists). Then deactivate(v), set $i \gets 0$, and go to H4.

H13. [Visit a solution.] If $e < n$, set $u \gets \mathtt{LLINK}[n]$ and $v \gets \mathtt{RLINK}[n]$; go to H14 if $\mathtt{NBR}[u]\,[v] = \infty$; otherwise set $\mathtt{EU}[e] \gets u$, $\mathtt{EV}[e] \gets v$, $e \gets n$. Now visit the n-cycle defined by arrays $\mathtt{EU}$ and $\mathtt{EV}$. (See exercise 113.)

H14. [Back up.] Terminate if $l = 0$. Otherwise set $l \gets l-1$.

H15. [Undo changes.] Set $d \gets \mathtt{D}(l)$ and $i \gets \mathtt{I}(l)+1$. Go to H14 if $i \ge d$. Otherwise set $e \gets \mathtt{E}(l)$, $k \gets (l > 0?\ \mathtt{A}(l-1)\colon 0)$, $a \gets \mathtt{A}(l)$, $v \gets n$. While $k < a$, set $u \gets \mathtt{ACTIVE}[k]$, $\mathtt{RLINK}[v] \gets u$, $\mathtt{LLINK}[u] \gets v$, $v \gets u$, $k \gets k+1$. Then set $\mathtt{RLINK}[v] \gets n$, $\mathtt{LLINK}[n] \gets v$, $\mathtt{S} \gets \mathtt{S}(l)$, $\mathtt{T} \gets \mathtt{T}(l)$. For $0 \le k < \mathtt{S}$, set $u \gets \mathtt{VIS}[k]$ and $\bigl(\mathtt{MATE}(u),\mathtt{DEG}(u)\bigr) \gets \mathtt{SAVE}[nl+u]$. Finally set $\mathtt{CURV} \gets \mathtt{CV}(l)$. Go to H5 if $l > 0$.

H16. [Advance at root level.] Terminate if $\mathtt{CURV} < 0$. Otherwise set $\mathtt{CURU} \gets \mathtt{MATE(CURV)}$. (The previous edge $\mathtt{CURU}$ — $\mathtt{CURV}$ is gone.) Set $\mathtt{LLINK}[n] \gets \mathtt{RLINK}[n] \gets n$, $a \gets 0$, $\mathtt{MATE(CURU)} \gets \mathtt{MATE(CURV)} \gets -1$, $\mathtt{S} \gets n$. (Everything is again bare.) If $\mathtt{DEG(CURU)} = 2$, set $\mathtt{TRIG[0]} \gets \mathtt{CURU}$ and $\mathtt{T} \gets 1$; otherwise set $\mathtt{T} \gets 0$. If $\mathtt{DEG(CURV)} = 2$, set $\mathtt{TRIG[T]} \gets \mathtt{CURV}$ and $\mathtt{T} \gets \mathtt{T}+1$. Go to H3 if $\mathtt{T} = 0$. Otherwise set $\mathtt{CV(0)} \gets -1$, $\mathtt{A(0)} \gets e \gets 0$, and go to H6. ▌

This marvelous algorithm has lots of steps, but it isn't terribly hard to understand. Its length arises mostly from the fact that a variety of data structures need to work together, combined with the fact that special provisions must be made at root level when no vertex has degree 2. In such cases, which are handled in steps H3 and H16, we choose a root vertex of minimum degree, and a root edge that touches it. We find all Hamiltonian cycles for which the root edge is present; then we discard that edge, and repeat the process. Eventually we will see a vertex of degree 2.

Table 1

A BAKER'S BAKER'S DOZEN OF EXAMPLE GRAPHS FOR ALGORITHM H

Graph	Description	Vertices	Edges	Degrees min . . max	Hamiltonian cycles	Running time (mems)	Mems per solution
A	*Anna Karenina*	49	138	3 . . 19	49152	415M	8447.1
B	binary trees	42	84	4 . . 4	14306485	8790M	614.4
C	concentric rings	144	216	3 . . 3	66770562	35G	528.5
D	disconnected	23	118	7 . . 22	0	66G	∞
E	expander	48	96	4 . . 4	107921396	70G	652.3
F	Fleischner G_3	57	126	3 . . 14	2	8872M	$4.4 \cdot 10^9$
G	giraffe tours	108	224	2 . . 8	4515918298	3217G	712.4
H	Halin from π	128	227	3 . . 11	10128654600	2929G	289.2
P	parity clash	101	182	2 . . 4	0	141T	∞
Q	5-cube	32	80	5 . . 5	906545760	249G	275.3
R	"random"	64	125	2 . . 6	9011601	7127M	790.9
S	Sierpiński simplex	34	96	3 . . 6	1165688832	311G	267.0
T	tripartite $K_{4,5,6}$	15	74	9 . . 11	207360000	40G	191.5
U	USA from `ME`	50	154	2 . . 48	68656026	182G	2651.2

One good way to start learning Algorithm H is to play through the steps by hand when G is a small graph like K_3 or K_4 or C_4. (See exercise 115.)

Algorithm H promises to produce interesting results galore, because the number of interesting graphs is enormous. We can get an idea of its performance in practice by studying the statistics of the benchmarks in Table 1, which reports on many different kinds of graphs. Each graph has been given an identifying letter for convenience.

- A, an "unstructured" graph based on Tolstoy's novel *Anna Karenina*. More precisely, this smallish graph arises from *book*(`"anna"`$,0,6,0,0,1,1,0$) in the Stanford GraphBase (SGB) after repeatedly removing vertices whose degree is less than 3. Algorithm H finds its Hamiltonian cycles in a flash.

- B, by contrast, is a 4-regular graph with a strict mathematical structure. It's the SGB graph *binary*$(5,5,0)$, which consists of the binary trees with 5 internal nodes; they're related by the "rotation" operation, 7.2.1.6–(12). (Algorithm 7.2.1.6L defines a Hamiltonian *path* in this graph, not a cycle.)

- C is one of the generalized dodecahedron graphs considered in exercise 11, with parameter $q = 36$ where Hamilton's original graph had $q = 5$.

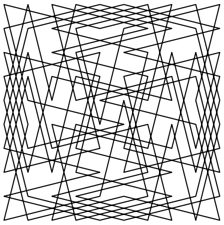
- D is a contrived example that's obtained from six disjoint copies of K_3, together with a copy of K_5 whose five vertices are joined to everything else. It has no Hamiltonian cycles, because it breaks into six nonempty components when the five vertices of K_5 are removed. (In the terminology of exercise 117, graph D isn't "tough." Every Hamiltonian graph is tough.)

- E is the SGB graph *raman*$(3,47,1,1)$, a "Ramanujan graph of type 1." The vertices are $\{0,1,\ldots,46,\infty\}$, and the edges are described in exercise 119.

tripartite graph, complete
benchmarks+
unstructured
Tolstoy
book
Stanford GraphBase
SGB
4-regular graph
regular graph
binary
binary trees
dodecahedron
concentric rings
Hamilton
components
tough
SGB
raman
Ramanujan graph
expander graph, see *raman*

- F is a minor of the amazing 338-vertex graph introduced by H. Fleischner in 2013. His graph has 318 vertices of degree 4, 20 vertices of degree 14, and exactly one Hamiltonian cycle(!). (See exercise 120.)

- G is a graph of 10×10 "giraffe" moves, where a giraffe is like a knight in chess except that it's a $(4,1)$-leaper instead of a $(2,1)$-leaper. For example, the attractive tour shown here, which has 90°-rotational symmetry, was published by Maurice Kraïtchik in §67 of his pioneering book *Le Problème du Cavalier* (Paris: Gauthiers[sic]-Villars, 1927). Graph G requires the giraffe to make a special kind of tour whose diagram exhibits a "Cossack cross" in the four central squares. (A symmetrical example of such a tour appears in the answer to exercise 888.)



- H is a representative example of a large family of planar Hamiltonian graphs called "Halin graphs." (See exercises 122–125.)

- P is a non-Hamiltonian graph that's another kind of Achilles heel for Algorithm H. We obtain it by appending new vertices '!' and '!!' to the 8×10 grid graph $P_8 \mathbin{\square} P_{10}$, with three new edges $u \,\text{---}\, ! \,\text{---}\, !! \,\text{---}\, v$, where u and v are opposite corners of the grid. There's no Hamiltonian cycle; for if we collapse '!' and '!!' into a single vertex, we obtain an equivalent bipartite graph P' with 41 vertices in one part and 40 vertices in the other. Unfortunately, Algorithm H doesn't understand this. So it explores zillions of fruitless paths.

- Q is the familiar 5-cube, $P_2 \mathbin{\square} P_2 \mathbin{\square} P_2 \mathbin{\square} P_2 \mathbin{\square} P_2$, whose Hamiltonian cycles are the 5-bit "Gray cycles" that we investigated in Section 7.2.1.1.

  Their total number, about 9 million, is just half of the value of $d(5)$ that was reported in Eq. 7.2.1.1–(26). Hmmm; was that a mistake? No: $d(5)$ considers the cycles $(x_0 \ldots x_{31})$ and $(x_{31} \ldots x_0)$ to be different, while Algorithm H does not.

- R is a graph obtained by adding 64 "random" edges to a 64-cycle. More precisely, it's the SGB graph whose official name is

$$\mathit{gunion}(\mathit{random\_graph}(64,64,0,0,0,0,0,1,1,3142),\mathit{board}(64,0,0,0,1,1,0),0,0).$$

It has only 125 edges, because three of the added edges were already present.

- S is the Sierpiński tetrahedron $S_3^{(4)}$, which was defined and illustrated in Section 7.2.2.3 (Fig. 114). Lots and lots of Hamiltonian cycles here.

- T is a special case of exercise 106.

- U, the graph that's last but not least in Table 1, is another unstructured example from "real life." It revisits the graph of the 48 contiguous states of the USA, 7.1.4–(133), augmented by two additional vertices '!' and '!!'; there are edges $!! \,\text{---}\, \texttt{ME}$, and $! \,\text{---}\, v$ for all $v \notin \{\texttt{ME}, !\}$. Thus its Hamiltonian cycles are the same as the Hamiltonian *paths* in 7.1.4–(133) from ME to any other state. (We used ZDD technology to treat those 68 million paths in Section 7.1.4.)

Fleischner
unique Hamiltonian cycle
giraffe
knight
leaper
90°-rotational symmetry
Kraïtchik
Cossack cross
cross
Halin graphs
grid graph
bipartite graph
5-cube
Gray cycles
random
gunion
random_graph
board
Sierpiński tetrahedron
unstructured
ZDD

A census of knight's tours.

Who knows what I might eventually decide to say next? Please stay tuned. [There will be an Algorithm B, analogous to Algorithm H, which visits all Hamiltonian cycles of bidirected graphs — which generalize digraphs.]

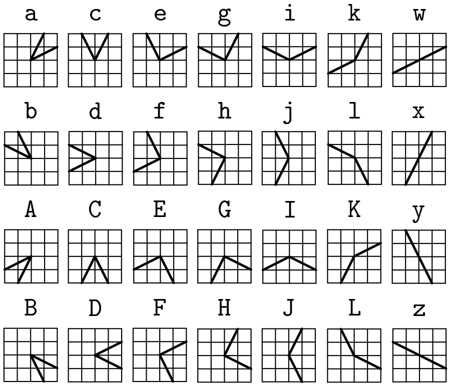
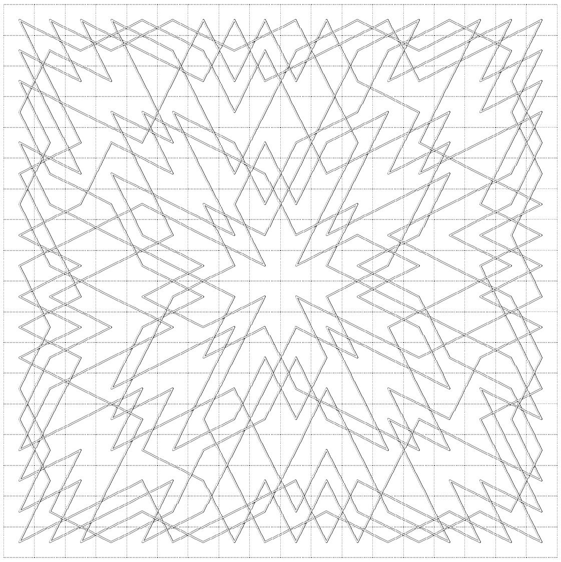
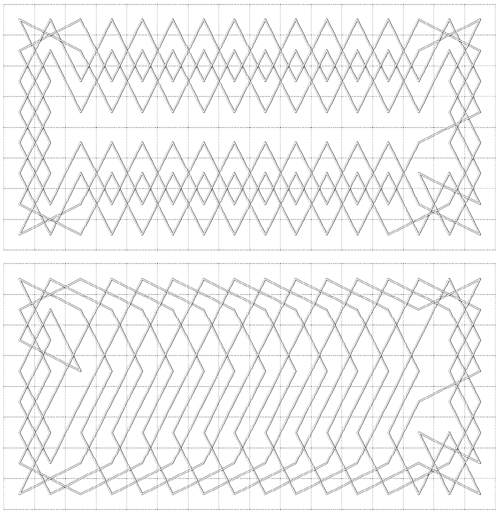


Fig. 123. The 28 possible wedges of a knight. The angle θ of the narrowest wedges, $\{\mathtt{a},\mathtt{b},\mathtt{A},\mathtt{B}\}$, is $\arctan\frac{3}{4} \approx 37^\circ$; $\{\mathtt{c},\mathtt{d},\mathtt{C},\mathtt{D}\}$ are slightly wider, $90^\circ - \theta = \arctan\frac{4}{3} \approx 53^\circ$. Eight wedges, namely $\{\mathtt{e},\mathtt{f},\mathtt{E},\mathtt{F},\mathtt{g},\mathtt{h},\mathtt{G},\mathtt{H}\}$, make a 90° turn. Then come $\{\mathtt{i},\mathtt{j},\mathtt{I},\mathtt{J}\}$, at $90^\circ + \theta \approx 127^\circ$; and $\{\mathtt{k},\mathtt{l},\mathtt{K},\mathtt{L}\}$, at $180^\circ - \theta \approx 143^\circ$. Finally, wedges $\{\mathtt{w},\mathtt{x},\mathtt{y},\mathtt{z}\}$ are straight. Notice that a 90° counterclockwise rotation changes $\mathtt{a} \mapsto \mathtt{b} \mapsto \mathtt{A} \mapsto \mathtt{B} \mapsto \mathtt{a}$; ...; $\mathtt{k} \mapsto \mathtt{l} \mapsto \mathtt{K} \mapsto \mathtt{L} \mapsto \mathtt{k}$; $\mathtt{w} \mapsto \mathtt{y} \mapsto \mathtt{w}$; and $\mathtt{x} \mapsto \mathtt{z} \mapsto \mathtt{x}$.

$$\alpha_2\alpha_3\alpha_4\alpha_1,\ \alpha_3\alpha_4\alpha_1\alpha_2,\ \alpha_4\alpha_1\alpha_2\alpha_3,\ \bar\alpha_4\bar\alpha_3\bar\alpha_2\bar\alpha_1,\ \bar\alpha_3\bar\alpha_2\bar\alpha_1\bar\alpha_4,\ \bar\alpha_2\bar\alpha_1\bar\alpha_4\bar\alpha_3,\ \bar\alpha_1\bar\alpha_4\bar\alpha_3\bar\alpha_2. \qquad (99)$$





History. There's something inherently satisfying about a path that "hits all the bases." For example, the ancient artist who carved the pattern of Fig. 124 in a mammoth's tusk might well have been trying to create a Hamiltonian-like path in an implicit grid graph, before the dawn of recorded history!

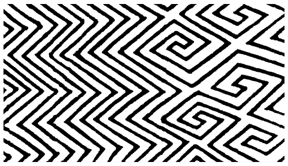


Fig. 124. A pattern from the Old Stone Age. This detail from a Paleolithic ivory carving, found near the present-day village of Mizyn in northern Ukraine, illustrates interesting ways to cover a grid, rotated 45°, with a nonintersecting path.

Of course we cannot read the minds of people who lived c. 15,000 B.C.; but we certainly can admire the wonderful sophistication that's evident in this fascinating artifact. [See M. Rudyns'kyj, *Industrie en os de la station paléolithique de Mizyn, interprétée par Fedir Vovk* (Kiev: Vseukraïns'ka Akademii͡a Nauk, Kabinet Antropolohiĭ im. F. Vovka, 1931).]

Fast forward now to 750 B.C., by which time "meander friezes" had become well developed in many cultures, especially in Greek pottery (see exercise 260).

A few hundred years later, another kind of Hamiltonian pattern appeared on icosahedra, as we saw near the beginning of this section. And considerable research on knight's tours began shortly after 800 A.D., as we've seen in (1) and (2).

But let's jump to the computer age. The first reasonably successful algorithm for visiting all Hamiltonian cycles of a given graph was published by S. M. Roberts and B. Flores in *CACM* **9** (1966), 690–694; **10** (1967), 201–202. They actually considered *directed* graphs; but of course their method also handled undirected graphs, because each edge $u \,—\, v$ can be represented by a pair of arcs, $\{u \longrightarrow v,\ v \longrightarrow u\}$. Their procedure was an early instance of straightforward backtracking: At each stage of the computation, a partial path $v_1 \longrightarrow \cdots \longrightarrow v_k$ was extended by trying all possible successors to v_k that hadn't yet appeared.

M. B. Wells, in §4.2.4 of his book *Elements of Combinatorial Computing* (1971), presented a similar method, but for the task of visiting all Hamiltonian *paths*, in *undirected* graphs. He explained how to detect certain impossible cases early in the search, by backtracking whenever a partial path $v_1 \,—\, \cdots \,—\, v_k$ wipes out all chances to reach a yet-untouched vertex v, namely when all of v's neighbors belong to $\{v_1, \ldots, v_k\}$. He also considered the untouched v that have exactly *one* neighbor in $\{v_1, \ldots, v_k\}$: There must not be three such vertices; and special restrictions apply when there are exactly two of them.

But G. R. Selby, in his Ph.D. thesis at Imperial College (University of London, 1970), 264 pages, realized that an edge-oriented approach would be much better. Instead of assigning vertices sequentially, he developed a "link algorithm" for undirected graphs that has much in common with Algorithm H above. Selby's method was not as symmetrical as it could have been — he still retained the concept of a "main" partial path $v_1 \,—\, \cdots \,—\, v_k$; but he augmented that path with a separate set of edges that it *implies*. Such edges, which form additional partial paths, arise when an untouched vertex belongs to only two available edges.

Historical notes
grid graph
Rudyns'kyj
Vovk
Ukraine
Stone Age
Paleolithic
Mizyn (Cyrillic 'Mezin')
meander friezes
Greek pottery
icosahedra
knight's tours
all Hamiltonian cycles
Roberts
Flores
directed graphs
backtracking
partial path
Wells
all Hamiltonian *paths*
undirected graphs
Selby
Imperial College
University of London
edge-oriented approach
link algorithm

Selby's co-advisor at Imperial College, N. Christofides, worked out a similar "multi-path algorithm" for *directed* graphs, and presented it in §10.2.3 of his book *Graph Theory: An Algorithmic Approach* (1975). Then S. Martello improved the algorithm further by incorporating the MRV heuristic, when selecting the initial vertex v_1 and when rank-ordering the neighbors of v_k that are candidates for v_{k+1}. [*ACM Trans. on Mathematical Software* **9** (1983), 131–138.] (The MRV heuristic had also been mentioned by Wells, who pointed out that neighbor ranking makes no difference to the total running time when we are visiting *all* of the solutions, because we are going to consider all choices of v_{k+1} anyway. However, just as with Warnsdorf's rule, MRV tends to find the *first* solution much faster.) Curiously none of these authors realized that it would be much better to use MRV symmetrically on the set of *all* endpoints of the current partial paths or directed paths, as in Algorithm H or Algorithm B, instead of always extending a "main" path by choosing a neighbor for v_k at every stage. (Selby did sometimes extend his main path at the left, if vertex v_1 had a forced neighbor v_0.)

F. Rubin [*JACM* **21** (1974), 576–580], unaware of Selby's or Wells's work, but inspired in part by S. L. Hakami [*IEEE Region Six Conf. Record* (1966), 635–643], proposed a similar algorithm that was in some ways weaker and in other ways stronger. His method repeatedly extended a single path at the right, while marking certain off-path edges as "required" and other edges as "deleted." (For example, edges that touch a vertex of degree 2 were "required.") But again, there was only a single main path. He also tested reachability between the path vertices and the remaining vertices; and he discussed preprocessing, whereby a graph could sometimes be partitioned into subgraphs whose Hamiltonian cycles could be pieced together to obtain the overall cycles.

W. Kocay [*Discrete Mathematics* **101** (1992), 171–188] realized that Selby and Christofides's multi-path algorithm could be made symmetrical, by giving equal status to every subpath. (Curiously, however, he still distinguished the left and right endpoints of subpaths. Instead of using MRV, the vertex that he chose for branching was a right endpoint of *maximum* degree(!) — see exercise 129.)

Kocay also went further, by backtracking when the current subproblem could not be completed to a Hamiltonian cycle because the current subgraph either had an articulation point or was bipartite in an impossible way. (See exercise 130.) This test was costly, but it could save considerable time in many cases.

Andrew Chalaturnyk (Master's thesis, University of Manitoba, 2008, vi + 123 pages) made Kocay's algorithm significantly faster by designing data structures that allow it to backtrack efficiently. He improved the method also by invoking tests for articulation points or bipartiteness only at judicious intervals, when chances for effective pruning of the search tree seemed most likely. Without such pruning (which was optional), his program was rather similar to Algorithm H, although considerably more complicated.

In unpublished experiments during 2001, D. E. Knuth had developed a symmetrical edge-based algorithm for Hamiltonian cycles in undirected graphs that was comparatively simple. He called it HAMDANCE, because it used data structures analogous to the dancing links of Algorithm 7.2.2.1X. Algorithm H,

Christofides
multi-path algorithm
Martello
MRV heuristic
Warnsdorf's rule
Rubin
Selby
Wells
Hakami
Kocay
articulation point
bipartite
Chalaturnyk
data structures
Knuth
HAMDANCE
data structures
dancing links

which uses sparse-set structures instead, was devised in 2024, and followed by Algorithm B in 2025.

sparse-set

EXERCISES

1. [*15*] We could save ourselves three syllables and three letters by saying “spanning cycle” and “spanning path” instead of “Hamiltonian cycle” and “Hamiltonian path.” Textbooks on graph theory could save lots of paper. Why doesn’t everybody do that?

▶ **2.** [*17*] Join every vertex of graph G to a new vertex, obtaining $G' = G \mathbin{—\!\!—} K_1$. True or false: G has a Hamiltonian path if and only if G' is Hamiltonian.

3. [*M22*] Reverse-engineer the rules by which Fig. 121’s vertices have been named.

4. [*M30*] The Hamiltonian cycle in Fig. 122(b) doesn’t look symmetrical. Show, however, that it has fourfold symmetry when drawn on an undistorted dodecahedron.

5. [*M20*] A *second* glance at the graph depicted in Fig. 122(c) reveals that it actually *is* obviously planar. Why?

6. [*22*] Draw the graph of the icosahedron in the style of Fig. 122(a), arranging the vertices in three concentric rings.

7. [*20*] Draw the graph of the 4-cube in the style of Fig. 122(c), using Gray binary code as the Hamiltonian cycle.

8. [*HM25*] Show that it’s possible to redraw the graph of the dodecahedron, Fig. 122, in such a way that all lines between adjacent vertices have the same length.

▶ **9.** [*M21*] A Hamiltonian cycle on a planar cubic graph, such as the dodecahedron in Fig. 121(b), can be described as a sequence of Ls and Rs denoting “left turn” and “right turn” at each vertex encountered during the cyclic journey.

a) Prove that no Hamiltonian cycle on the dodecahedron can contain any of the following subsequences: (i) LLLL; (ii) LRRL; (iii) LRLRLRL; (iv) LLRLRLL; (v) LLRLRR; (vi) LLRLL; (vii)–(xii), subsequences (i)–(vi) with L ↔ R swapped.

b) Therefore there is essentially only one cycle (and its dual obtained by L ↔ R).

10. [*24*] For which vertices v of Fig. 122(a) is there a Hamiltonian path from 12 to v?

11. [*M32*] The *generalized Petersen graph* $\mathrm{GP}(n,k)$ is an interesting cubic graph with $2n$ vertices $\{0,1,\ldots,n-1,0',1',\ldots,(n-1)'\}$ and $3n$ edges

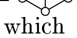
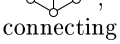
$$\{i \mathbin{—\!\!—} (i+1) \bmod n,\ i \mathbin{—\!\!—} i',\ i' \mathbin{—\!\!—} (i+k)' \bmod n \mid 0 \le i < n\}.$$

Figure 122(a) is the special case $q = 5$ of a general *concentric-ring graph* $\mathrm{GP}(2q,2)$.

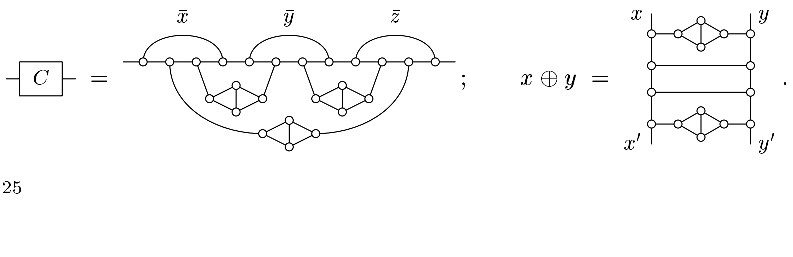
For which vertices v does the graph $\mathrm{GP}(2q,2)$ have a Hamiltonian path from $0'$ to v?

12. [*HM28*] How many Hamiltonian cycles exist in the graphs $\mathrm{GP}(2q,2)$?

14. [*22*] The one-in-three satisfiability problem of exercise 7.2.2.2–517 is NP-complete. For every such problem F, we shall construct a cubic graph G that is Hamiltonian if and only if F is satisfiable. Every edge of G corresponds to a Boolean variable; values of the variables for which the true edges form a Hamiltonian cycle will be called a *win*.

a) A cubic graph that contains $K_{2,1,1} =$  as an induced subgraph also contains the “Wheatstone bridge” , which has two edges that connect to other vertices. Show that those connecting edges must be true in every win.

b) For every clause $C = (x \vee y \vee z)$ of F, where x, y, and z are literals, include the “clause gadget” $-\!\boxed{C}\!-$ below as part of G. Show that $x + y + z = 1$ in every win.



spanning
Hamiltonian
fourfold symmetry
symmetry
planar
icosahedron
concentric rings
4-cube
Gray binary code
draw the graph
unit-distance graph
graph drawing
dodecahedron
Hamiltonian path
generalized Petersen graph
Petersen graph
$\mathrm{GP}(n,k)$
concentric-ring graph
enumeration of Ham cycles
one-in-three satisfiability problem
NP-complete
cubic graph
satisfiable
win
$K_{2,1,1}$
induced subgraph
Wheatstone bridge
literals
clause gadget

c) If two edges x and y of G are replaced by the "XOR gadget" $x \oplus y$ above, show that $x = x'$, $y = y'$, and $x = \bar{y}$ in every win.

d) Suppose the clauses of F are $\{C_1, \ldots, C_m\}$. Use the gadgets above to construct the desired graph G, starting with $-\!\boxed{C_1}\!-\cdots-\!\boxed{C_m}\!-$.

16. [*29*] What's the smallest connected cubic graph that is *not* Hamiltonian?

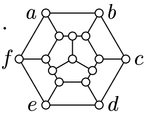
18. [*M20*] True or false: If a planar graph has a Hamiltonian cycle, so does its dual.

▶ **20.** [*M30*] (T. P. Kirkman, 1856.) Let G be a planar graph with n vertices and with exactly α_k k-sided faces for $k \geq 3$ (including the unbounded exterior face). For example, the graph of the dodecahedron, Fig. 122, has $n = 20$ and $\alpha_k = 12[k=5]$.

a) If G is Hamiltonian, prove that integers a_k exist such that $0 \leq a_k \leq \alpha_k$ and $\sum_{k=3}^{n}(k-2)a_k = n-2$. (For example, the dodecahedron has $a_k = 6[k=5]$.)

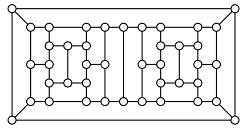
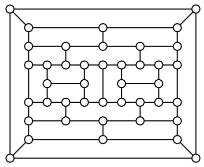
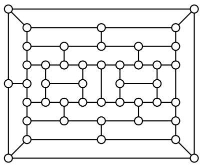
b) In a similar way, prove that the dodecahedron has no cycle of length 19.

c) Furthermore its vertices can't be completely covered by two *disjoint* cycles.



d) Use (a) to prove that every Hamiltonian cycle in the planar 16-vertex cubic graph G shown here must include the edges $a — b$, $c — d$, $e — f$.

e) Use (a) to decide whether any of the following graphs are Hamiltonian:

(i)  ; (ii)  ; (iii)  .

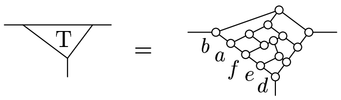
21. [*M25*] Large graphs that contain no Hamiltonian cycles can often be useful benchmarks. Construct infinitely many cubic planar graphs that fail to satisfy exercise 20(a).

24. [*M28*] A cubic graph is called *perfectly Hamiltonian* if its edges can be 3-colored in such a way that the edges of any two colors form a Hamiltonian cycle.

a) Which of the cubic graphs on 8 vertices are Hamiltonian? Perfectly Hamiltonian?

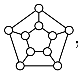
b) Prove that a *planar* cubic graph can be perfectly Hamiltonian only if there are nonnegative integers (a_k, b_k, c_k, d_k) for all $k \geq 3$ such that $a_k + b_k + c_k + d_k = \alpha_k$ is the number of k-faces as in exercise 20(a), and $\sum_k(k-2)a_k = \sum_k(k-2)b_k = \sum_k(k-2)c_k = \sum_k(k-2)d_k = (n-2)/2$, where n is the number of vertices.

27. [*M21*] The *Tutte gadget* is a useful 15-vertex graph fragment

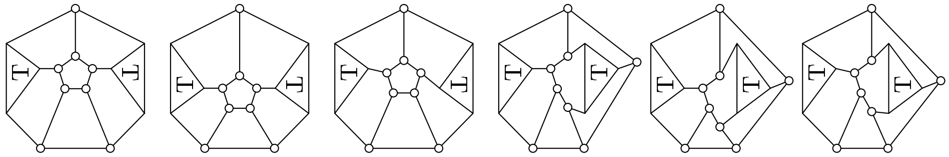


that can be obtained by removing vertex c from the 16-vertex graph in exercise 20(d).

a) Prove that every Hamiltonian path in a graph that contains the gadget must use the edge at the bottom of the T.

b) Prove that no Hamiltonian path in the pentagonal prism GP(5, 1), , includes the edges of two nonconsecutive "spokes."

c) Therefore none of the following 38-vertex graphs are Hamiltonian:



XOR gadget
gadgets
planar graph
dual
Kirkman
planar graph
faces
dodecahedron
cycle cover
benchmarks
perfectly Hamiltonian
edge coloring
cubic graphs
Tutte gadget
pentagonal prism
generalized Petersen graphs

d) Are any two of those six graphs isomorphic to each other?

30. [*20*] Each letter in a Græco-Roman icosahedron can be placed three ways within its triangular face, depending on the choice of “bottom edge” (except that Δ and O are symmetric). From this standpoint, the fact that Π and Y share the *same* bottom edge, in the text’s example from the British Museum, is a bit disconcerting.

Redesign that layout for the 21st century, so that (i) Roman letters `A`, `B`, ..., `T` replace the Greek ones; (ii) the bottom edge of a letter’s successor is always the upper left or upper right edge of the current letter; and (iii) `T` is adjacent to `A`, completing a cycle.

▶ **33.** [*M20*] Suppose G is an n-vertex graph that has H Hamiltonian cycles and h Hamiltonian paths that aren’t cycles. (Thus, there are H sets of n edges whose union is a cycle, and h sets of $n-1$ edges whose union is a path but not a cycle.) Let $G' = \{*\} \mathbin{—} G$ be the $(n+1)$-vertex graph obtained from G by adjoining a new vertex that’s adjacent to all the others. How many Hamiltonian cycles does G' have?

35. [*M25*] A close look at (1) shows that al-‘Adlī’s closed tour is traced by a “thread” that weaves alternately over and under itself at each crossing, forming a “knot.”

a) Prove that *every* closed knight’s tour can be drawn as such a knot.

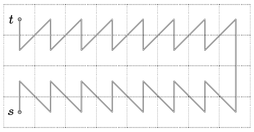
b) On the other hand, the over-under rule is violated four times in Ibn Manī‘’s open tour. Prove that every drawing of his tour must necessarily have at least four such exceptions to the rule.

36. [*22*] Find 4 × 8 knight’s tours that (i) preserve the syllables of Rudraṭa’s sloka, but differ from (2); (ii) preserve the fractured English syllables of (4).

37. [*22*] How many 4 × 8 knight’s tours are possible?

38. [*25*] Write a two-verse English poem for Rudraṭa’s 4 × 8 tour, analogous to (6).

40. [*25*] The variant of Chaturanga played in Rudraṭa’s day used a curious piece called an *elephant* (gaja) instead of a chess bishop. This piece had only five moves: One step forward or one step diagonally, representing the elephant’s trunk and its four legs. For example, an elephant can tour a 4×8 board by following the (s,t)-path illustrated here.



Represent this half-tour with a two-verse poem in English.

▶ **41.** [*M32*] This exercise classifies all elephant’s tours on an $m\times n$ board, for $m,n \ge 2$.

a) Let E_{mn} be the $m \times n$ elephant digraph. How many arcs does it have?

b) Does E_{mn} have a *closed* tour (a Hamiltonian *cycle*), for some values of m and n?

c) The open elephant’s tour in exercise 40 begins at the bottom left corner of E_{48}. Show that there’s also an open tour that begins at the *top* left corner of E_{48}.

d) Prove that every elephant’s tour must begin or end in the top row, when $m > 2$.

e) Similarly, prove that every such tour must begin or end in the bottom row.

f) Characterize all $m \times n$ elephant’s tours that begin in the top row.

g) Characterize all $m \times n$ elephant’s tours that begin in the bottom row.

h) Explain how to compute the number of Hamiltonian paths of E_{mn} that begin at a given vertex s and end at a given vertex t.

42. [*30*] Find a cycle of elephant moves on the 8 × 8 chessboard that (i) visits all but two of the cells, and (ii) has the fewest “trunk moves” among all such cycles.

44. [*18*] Using the syllables (7), construct a knight’s tour quatrain that *rhymes.*

46. [*19*] Draw Someshvara’s tour (8) in the style of (1).

50. [*19*] The text describes only one scenario for moves **9**, **10**, ..., that might extend the partial tour (10). What other paths are consistent with Warnsdorf’s rule?

isomorphic
Græco-Roman icosahedron
British Museum
Hamiltonian paths that aren’t cycles
al-‘Adlī
path diagrams
weaves
knot
alternating knot diagrams
Ibn Manī‘
Rudraṭa
sloka
fractured English
poem
Chaturanga
elephant
(s,t)-path: A path from vertex s to vertex t.
elephant’s tours
elephant digraph
trunk moves
quatrain
nonsense verse
Someshvara
path diagrams
Warnsdorf’s rule

51. [*21*] What paths does Algorithm W construct when g is the graph of knight moves on a 5×5 board, s is cell 00, $r = 1$, and t_1 is cell 44 (the corner opposite 00)?

52. [*20*] What is the behavior of Algorithm W if $t_i = t_j$ for some $i \neq j$?

▶ **53.** [*M21*] *Randomize* Algorithm W, by changing step W5 so that each candidate u is chosen with probability $1/q$ when there's a q-way tie for the minimum number of exits. *Hint:* There's a nice way to do this "on the fly" without building a table of candidates.

55. [*20*] How many of the 63 moves in the historic knight's tour (1) by Ibn Manī' agree with Warnsdorf's rule? Consider also the closed tour of al-'Adlī, with the same opening moves v_1 and v_2, as well as the open tour of Someshvara in exercise 46.

56. [*20*] Algorithm W sometimes moves to a "dead end" vertex (from which there's no exit), even though it could prolong the path by making a different choice. Discuss.

▶ **57.** [*21*] Design an algorithm to compute the tree of all possible paths that might be computed by Algorithm W, given g, s, and $\{t_1, \ldots, t_r\}$. Also compute, for each path, the probability that it would be obtained by the algorithm of exercise 53.

59. [*21*] What are the longest and shortest paths obtainable in the 8×8 knight graph when the *anti-Warnsdorf* rule is used? (Move always to a cell with the *most* exits.) Compare those results to the behavior of Algorithm W.

60. [*22*] Study empirically the behavior of Algorithm W on the concentric-ring graphs $R_q = \mathrm{GP}(2q, 2)$ of exercise 11, for $6 \le q \le 10$. What is the probability of obtaining (a) a Hamiltonian path? (b) a Hamiltonian cycle, when no target vertex is specified? (c) a Hamiltonian cycle, when there's a single target vertex with $s \mathrel{-\!\!-} t_1$?

62. [*16*] Prove that Algorithm W always finds a Hamiltonian path when $g = P_m \mathbin{\square} P_n$ is the $m \times n$ grid graph, $s = (0, 0)$ is a corner vertex, and $r = 0$.

▶ **63.** [*M30*] Prove that Algorithm W always finds a Hamiltonian path in the special case when $g = P_2 \mathbin{\square} P_2 \mathbin{\square} \cdots \mathbin{\square} P_2$ is an n-cube and $r = 0$.

▶ **65.** [*25*] Is there a Hamiltonian graph for which Algorithm W always *fails* to find a Hamiltonian path, regardless of the starting point and the ordering of arcs?

70. [*11*] Show that step F4 sometimes calls '$\mathit{update}(u_1, \ldots, u_t)$', which does nothing.

71. [*M20*] Euler believed that his method for discovering tours was "safe" and "infallible"; but (16) is a case where it fails to find a cycle. Construct arbitrarily large Hamiltonian graphs for which Algorithm F can in fact get stuck with paths of length 10.

73. [*21*] Discuss implementing the dictionary of Algorithm F with a hash table based on linked lists. If the entry for each path links to the number of the previous path that belongs to the same list, step F6 can regard all links $\le p_2$ as null.

75. [*M23*] (*One-sided flips.*) Simplify Algorithm F so that it flips subpaths only at the right, and doesn't distinguish between paths and cycles; call the resulting procedure "Algorithm F$^-$." (More precisely, Algorithm F$^-$ never goes to step F5; it omits the second *update* in step F4; and it doesn't put paths into canonical form.)

a) Suppose Algorithm F$^-$ is applied to a Hamiltonian path $v_1 \mathrel{-\!\!-} \cdots \mathrel{-\!\!-} v_n$ in a *cubic* graph G. Show that it constructs a *cycle* of Hamiltonian paths, each beginning with v_1, and illustrate this cycle when G is the 3-cube.

b) Furthermore the number of *cyclic* Hamiltonian paths in that cycle is even.

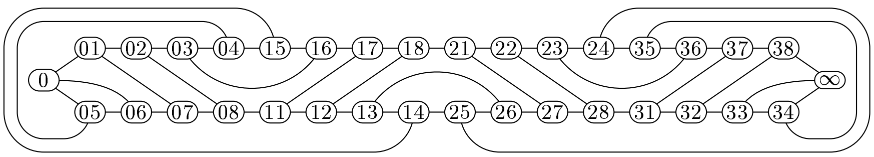
c) Moreover, the number of Hamiltonian cycles containing any given edge is even.

d) Every cubic Hamiltonian graph therefore has at least three Hamiltonian cycles.

e) Every cubic graph with exactly three Hamiltonian cycles is *perfectly* Hamiltonian.

Randomize
Ibn Manī'
al-'Adlī
Someshvara
anti-Warnsdorf
$m \times n$ knight graph: The SGB graph $\mathit{board}(m$
concentric-ring graphs
generalized Petersen graphs
Warnsdorf's rule
grid graph
n-cube
$\mathit{update}(u_1, \ldots, u_t)$
Euler
hash table
linked lists
null
One-sided flips
canonical form
lollipop method, see one-sided flips
cubic
3-cube
perfectly

77. [*M26*] The *Cameron graph* C_n *of order* n is a planar cubic graph on the $8n+2$ vertices $\{ij \mid 0 \le i < n, 1 \le j \le 8\} \cup \{0, \infty\}$ defined by the relations $i7 — i1 — i2 — i3 — i4 — i5 — i6 — i7 — i8 — i2$, $i3 — (i{+}1)6$, $i4 — (i{+}1)5$, and $i8 — (i{+}1)1$ for all integers i; replace all vertices ij for $i < 0$ by 0, and all ij for $i \ge n$ by ∞. For example,

$C_4 =$


.

a) Prove that the involution $ij \leftrightarrow (n{-}1{-}i)(9{-}j)$ is an automorphism of C_n.

b) Prove that C_n has exactly three Hamiltonian cycles (one of which is the "obvious" cycle $\alpha_n = 0 — 01 — 02 — 03 — \cdots — \infty — \cdots — 07 — 06 — 05 — 0$).

c) Compute the number c_n of one-sided flips needed to go from α_n to its mate β_n with respect to $0 — 01$, in the sense of answer 75(c), for $1 \le n \le 9$.

d) Surprise! Exactly $c_{n-2} + 10$ flips go from α_n to β_n with respect to $01 — 0$.

e) How many flips go from α_n to its mate γ_n with respect to (i) $0 — 05$? (ii) $05 — 0$?

78. [*22*] Study empirically the behavior of Algorithm F on the concentric-ring graphs $R_q = \mathrm{GP}(2q, 2)$ of exercise 11, for $6 \le q \le 10$ and $q = 100$. Let $t = 1$, and choose v_1 at random; also randomize the order in which a vertex's neighbors are examined. Estimate the probability of obtaining (a) a Hamiltonian path; (b) a Hamiltonian cycle. How many nontrivial calls of *update* are typically needed, before succeeding?

79. [*M32*] (N. Beluhov, 2019.) Say that two Hamiltonian paths or cycles are *equivalent* if they can be transformed into each other by Algorithm F.

a) Find a graph with two inequivalent cycles.

b) Can a graph have arbitrarily many pairwise inequivalent cycles?

80. [*M20*] For which $q_1, \ldots, q_s, t$ is the graph $(K_{q_1} \oplus \cdots \oplus K_{q_s}) — K_t$ Hamiltonian?

81. [*M27*] (*Forcibly Hamiltonian degrees.*) Sometimes we can conclude that a graph is Hamiltonian just by knowing that it has lots of edges. If $n > 2$ and the vertices of G have respective degrees $d_1 \le d_2 \le \cdots \le d_n$, we shall prove that G is Hamiltonian whenever

$$1 \le k < n/2 \quad \text{and} \quad d_k \le k \quad \text{implies} \quad d_{n-k} \ge n-k. \qquad (*)$$

a) If G satisfies $(*)$ and has $m < \binom{n}{2}$ edges, so that G is not the complete graph K_n, prove that G contains two nonadjacent vertices $\{u, v\}$ with $\deg(u) + \deg(v) \ge n$.

b) Continuing (a), let $G_0 = G$; and let $G_{k+1} = G_k \cup \{u_k — v_k\}$, where $u_k \not— v_k$ and $\deg(u_k) + \deg(v_k) \ge n$ in G_k, for $0 \le k < \binom{n}{2} - m$. Explain how to construct a Hamiltonian cycle in G_k, given a Hamiltonian cycle in G_{k+1}. (Since $G_{\binom{n}{2}-m} = K_n$ is Hamiltonian, so too is G_0.) *Hint:* Use flips as in Algorithm F.

82. [*M25*] If condition $(*)$ fails in G for at least one value of k, show that there's a non-Hamiltonian graph G' whose degree sequence $d'_1 \le d'_2 \le \cdots \le d'_n$ satisfies $d_1 \le d'_1$, $d_2 \le d'_2$, …, $d_n \le d'_n$. (In this sense exercise 81 is the best possible result of its kind.)

83. [*M30*] (C. S. A. Nash-Williams.) Let G be an r-regular graph with $2r+1$ vertices.

a) Prove that r is even.

b) Prove that G has a Hamiltonian path $u_0 — u_1 — \cdots — u_{2r}$.

c) If G isn't Hamiltonian, show that $u_0 — u_j \iff u_{j-1} \not— u_{2r}$, for $1 < j < 2r$.

d) If G isn't Hamiltonian, show that it has a cycle $v_1 — v_2 — \cdots — v_{2r} — v_1$.

e) Conclude that G *is* Hamiltonian. *Hint:* Suppose $v_0 — v_j \iff j$ is odd.

Cameron graph
planar cubic graph
involution: perm of order 2
flips
mate
concentric-ring graphs
generalized Petersen graphs
Beluhov
equivalent
Forcibly Hamiltonian degrees
degree seq of graph
complete graph
flips
degree sequence
Nash-Williams
r-regular

84. [*M28*] What's the smallest number of edges for which condition $(*)$ in exercise 81 forces an n-vertex graph to be Hamiltonian?

85. [*HM21*] (P. Erdős, 1962.) Let $f(n,k) = \binom{n-k}{2} + k^2$, and $g(n,k) = \max(f(n,k), f(n, \lfloor (n-1)/2 \rfloor))$. Prove that if $1 \le k < n/2$, there's a non-Hamiltonian graph with $g(n,k)$ edges and n vertices, where every vertex has degree $\ge k$. But every such graph with *more* than $g(n,k)$ edges is Hamiltonian. *Hint:* When is $f(n,k) \ge f(n,k+1)$?

▶ **86.** [*HM25*] A graph is called *traceable* if it has a Hamiltonian path. Continuing exercise 85, determine the largest possible number of edges in a nontraceable n-vertex graph for which the degree of every vertex is k or more. *Hint:* Let the function $\hat{f}(n,k) = \binom{n-1-k}{2} + k(k+1)$ play the role of $f(n,k)$ in that exercise.

88. [*M27*] The *length* of a graph is the number of edges in its longest path. (For example, the 4×4 knight graph has length 14.)

a) Let G be a connected graph whose n vertices each have degree k or more, where $k < n/2$. Prove constructively that the length of G is at least $2k$.

b) Prove that an n-vertex graph of length l has at most $nl/2$ edges.

c) Exhibit an n-vertex graph of length l and at least $nl/2 - (l+1)^2/8$ edges.

▶ **89.** [*M31*] The *circumference* of a graph is the number of edges in its longest cycle. (For example, the 4×4 knight graph has circumference 14.)

a) Let G be a biconnected graph whose n vertices each have degree k or more, where $1 < k \le n/2$. Prove constructively that the circumference of G is at least $2k$.

b) Prove that an n-vertex graph of circumference c has at most $(n-1)c/2$ edges.

c) If $c>2$, exhibit an n-vertex graph of circumference c and $\ge nc/2 - (c+1)^2/8$ edges.

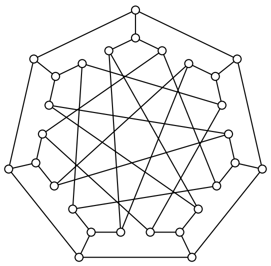
90. [*16*] True or false: The length of G is two less than the circumference of $K_1 \mathbin{—} G$.

93. [*M25*] (J. W. Moon, 1965.) A graph that has a Hamiltonian path between every pair of vertices $u \ne v$ is called *Hamiltonian connected.*

a) True or false: Every vertex of a Hamiltonian-connected graph has degree ≥ 3.

b) Construct a Hamiltonian-connected graph on $n \ge 4$ vertices that has the smallest possible number of edges (for example, 8 edges when $n = 5$; 9 edges when $n = 6$).

▶ **95.** [*M28*] The *Coxeter graph* is a remarkable cubic graph whose 28 vertices $\{a_j, b_j, c_j, d_j \mid 0 \le j < 7\}$ are connected by the edges $a_j \mathbin{—} d_j$, $b_j \mathbin{—} d_j$, $c_j \mathbin{—} d_j$, $a_j \mathbin{—} a_{j+1}$, $b_j \mathbin{—} b_{j+2}$, $c_j \mathbin{—} c_{j+3}$, for $0 \le j < 7$. (All subscripts are treated modulo 7. Vertices $a_0, \ldots, a_6$ form the "outer ring" of the illustration.)



a) Determine its automorphisms, by finding a Sims table as in Section 7.2.1.2. (Use the ordering $(a_6, b_6, c_6, d_6, \ldots, a_0, b_0, c_0, d_0)$; thus, for example, the permutations of $S_{n-2} = S_{26}$ will fix the final vertex d_0.) *Hint:* There will be a surprise!

b) Show that it is a *vertex-transitive graph*: Given any vertices v and v', there's an automorphism that takes $v \mapsto v'$. ("All vertices are alike.")

c) Show that it's also an *edge-transitive graph*: Given any edges $u \mathbin{—} v$ and $u' \mathbin{—} v'$, there's an automorphism that takes $\{u,v\}$ into $\{u',v'\}$. ("All edges are alike.")

d) Furthermore, it's a *hypohamiltonian graph*: It has no Hamiltonian cycle, yet it does become Hamiltonian when any vertex is removed.

▶ **100.** [*HM30*] Analyze the cycle covers of the flower snark graph J_q, for $q > 2$ (see exercise 7.2.2.2–176). How many of them have exactly k cycles?

▶ **103.** [*M25*] If a graph G has a Hamiltonian cycle H, show that there's a very easy way to test whether or not G is planar. *Hint:* See Algorithm 7B.

Erdős
traceable
nontraceable
length
knight graph
circumference
biconnected graph
Moon
Hamiltonian connected
Coxeter graph
cubic graph
automorphisms
Sims table
vertex-transitive graph
edge-transitive graph
hypohamiltonian graph
cycle covers
flower snark graph
planar

105. [*M18*] Exactly how many Hamiltonian cycles are present in (a) the complete graph K_n? (b) the complete bipartite graph $K_{m,n}$?

106. [*M26*] Continuing exercise 105, enumerate the Hamiltonian cycles of $K_{l,m,n}$.

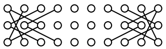
▶ **108.** [*24*] According to the discussion in the text, every Hamiltonian cycle that contains edge $\genfrac{}{}{0pt}{}{02}{21}$ of the 3 × 10 knight graph also contains the edge $\genfrac{}{}{0pt}{}{01}{13}$.

a) Given those edges, show that either $\genfrac{}{}{0pt}{}{07}{28}$ or $\genfrac{}{}{0pt}{}{16}{28}$ must be chosen.

b) And if that choice is $\genfrac{}{}{0pt}{}{16}{28}$, another edge is forced.

c) Continuing (b), show that edge $\genfrac{}{}{0pt}{}{03}{22}$ leads to a contradiction.

d) Continuing (c), consider the consequences of now choosing $\genfrac{}{}{0pt}{}{03}{15}$.

109. [*15*] After Algorithm H deduces the starting pattern  for the 3×10 knight graph, what are the values of `MATE(0)`, `MATE(1)`, `MATE(2)`, `MATE(3)`, `MATE(4)`?

111. [*23*] How much space should be allocated for the arrays `TRIG`, `ACTIVE`, and `SAVE` in Algorithm H so that no memory bounds will be exceeded?

▶ **112.** [*26*] Exactly what changes to the data structures should be made in step H8 of Algorithm H when we have (a) `MATE(`u`)` < 0 and `MATE(`w`)` < 0? (b) `MATE(`u`)` < 0 and `MATE(`w`)` ≥ 0? (c) `MATE(`u`)` ≥ 0 and `MATE(`w`)` < 0? (d) `MATE(`u`)` ≥ 0 and `MATE(`w`)` ≥ 0?

113. [*20*] Design an algorithm to "unscramble" the cycle defined by arrays `EU` and `EV` in step H13: It should find a permutation such that $v_1 — v_2 — \cdots — v_n — v_1$.

115. [*M23*] Describe the search tree of Algorithm H when G is the complete graph K_n.

116. [*20*] Find a Hamiltonian cycle in the graph *binary*(4, 4, 0) (see Table 7.2.1.6–3).

117. [*M22*] (V. Chvátal, 1973.) The *toughness* $t(G)$ of graph G is $\min |U|/k(G\backslash U)$, where the minimum is taken over all sets of vertices U such that $G \setminus U$ is disconnected, and k denotes the number of components. (If G is the complete graph K_n, no set U disconnects it, and we have $t(K_n) = \infty$.) We say that G is "tough" if $t(G) \geq 1$.

a) True or false: $t(G) = 0$ if and only if G isn't connected.

b) Show that $t(G \setminus e) \leq t(G)$ for all edges e.

c) Prove that every Hamiltonian graph is tough.

d) Evaluate $t(K_{m,n})$ when $m \leq n$.

e) What's $t(G)$ when G is the Petersen graph (which isn't Hamiltonian)?

118. [*M27*] Continuing exercise 117, determine $t(K_m \,\square\, K_n)$, when $m, n > 1$.

119. [*M24*] Read the SGB source code of *raman* and explain the edges of graph E.

120. [*M22*] Let G_0 be the graph with vertices $\{i, j, k, u, v, w, x, y, z, U, V, W, X, Y, Z\}$ and the following 24 edges: $i — j — k — i$; $v — i — V$, $w — j — W$, $x — k — X$; $u — U$, $v — V$, $w — W$, $x — X$; $u — y — z$, $U — Y — Z$; $u — v$, $U — X$; $y — w$, $Y — W$; $x — z — Z — V$.

a) Prove that G_0 has four automorphisms and exactly two Hamiltonian cycles.

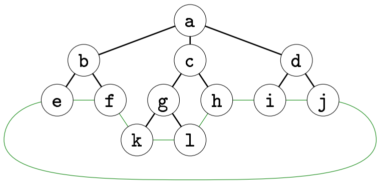
b) Let $(l_1, l_2, l_3) = (i, j, k)$; $(q_1, q_2, q_3) = (v, w, x)$; $(Q_1, Q_2, Q_3) = (V, W, X)$; and $(r_1, r_2, r_3) = (u, y, z)$. For $1 \leq t \leq 3$, let $G_0^{(t)}$ be a copy of G_0 with vertices $i^{(t)}$, …, $Z^{(t)}$. Obtain graph G_t by appending $G_0^{(t)}$ and then doing this: (i) remove vertex q_t; (ii) remove $u^{(t)} — U^{(t)}$ and the edge $q_t — Q_t$; (iii) replace the edges $l_t — q_t$ and $r_t — q_t$ by $l_t — u^{(t)}$ and $r_t — U^{(t)}$; (iv) add edges from Q_t to all the vertices of $G_0^{(t)}$ except $i^{(t)}$, $j^{(t)}$, $k^{(t)}$. (Graph G_t therefore has $15 + 14t$ vertices and $24 + 34t$ edges.) Prove that G_t has exactly two Hamiltonian cycles.

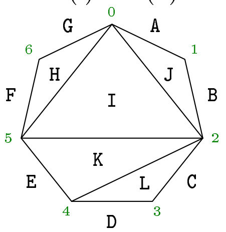
▶ **121.** [*21*] Find all the 10 × 10 giraffe tours that are symmetric under 90° rotation.

complete graph
K_n
complete bipartite graph
$K_{m,n}$
$K_{l,m,n}$
tripartite graph, complete
3×10 knight graph
`MATE`
`TRIG`
`ACTIVE`
`SAVE`
memory bounds
unscramble
complete graph K_n
binary
Chvátal
toughness
disconnected
cutsets: sets of vertices that disconnect a graph
components
complete graph
K_n
$K_{m,n}$
Petersen graph
rook graph $K_m \,\square\, K_n$
SGB
raman
automorphisms
giraffe tours

▶ **122.** [*M27*] A *Halin graph* can be defined in two complementary ways: (i) Let T be a tree in which no node has degree 1, and the root has degree ≥ 3. Let the leaves of T be $x_0x_1\ldots x_{q-1}$ in preorder, and let C be the cycle $x_0 — x_1 — \cdots — x_{q-1} — x_0$. Then $H = T \cup C$ is a Halin graph. (ii) Let C be a regular q-gon with vertices 0, 1, …, $q-1$. Let $\{i_1 — j_1,\ \ldots,\ i_t — j_t\}$ be nonintersecting chords of C; they partition the interior of C into $t+1$ regions. Let H be the graph of order $q+t+1$ whose vertices are the sides of C, together with those regions. Two sides are adjacent in H if they are consecutive; a region is adjacent to the sides on its boundary and to the regions with which it shares a boundary. Then H is a Halin graph. Notice that, under either definition, a Halin graph must be planar, and each of its vertices must have degree 3 or more.

For example, here are structures that respectively illustrate (i) and (ii) with $q = 7$:





a) Prove that the corresponding graphs are isomorphic, by finding a correspondence between vertices $\{\mathtt{a},\mathtt{b},\ldots,\mathtt{l}\}$ and vertices $\{\mathtt{A},\mathtt{B},\ldots,\mathtt{L}\}$ that preserves adjacency.

b) If H satisfies definition (i), prove that it also satisfies definition (ii).

c) If H satisfies definition (ii), prove that it also satisfies definition (i).

123. [*M30*] How many nonisomorphic Halin graphs have n vertices, for $n \leq 1000$?

124. [*M25*] A graph is *uniformly Hamiltonian* if, for every edge e, it contains a Hamiltonian cycle C^+ with $e \in C^+$ as well as a Hamiltonian cycle C^- with $e \notin C^-$. Prove that every Halin graph is uniformly Hamiltonian.

125. [*20*] Use the decimal digits $(\pi_0.\pi_1\pi_2\ldots)_{10}$ of π to define t nonintersecting chords $(i_1 — j_1,\ \ldots,\ i_t — j_t)$ of a regular 100-gon for $0 \leq t < 98$, by letting $i_k = \pi_{4r}\pi_{4r+1}$ and $j_k = \pi_{4r+2}\pi_{4r+3}$, where r is as small as possible with respect to previous chords. For example, the sequence begins (31 — 41, 59 — 26, 53 — 58, 97 — 93, 23 — 84, 62 — 64, …); the next chord cannot be 33 — 83, because that one overlaps 31 — 41. These chords define Halin graphs $H_\pi^{(t)}$ with $101+t$ vertices, by exercise 122.

What is the final chord, $i_{97} — j_{97}$?

127. [*20*] There's obviously no Hamiltonian path in the graph 7.1.4–(133) from `ME` to any of the other states of New England (`NH`, `VT`, `MA`, `CT`, `RI`), because `NY` is an articulation point. Is there a Hamiltonian path from `ME` to every *other* state?

▶ **128.** [*20*] Which of the 14 benchmark graphs in Table 1 are planar?

▶ **129.** [*24*] The MRV heuristic used in step H11 to choose a vertex for branching prefers small degree d, because the search tree has a d-way branch. On the other hand, one can argue that large d is actually good, because step H12 removes d edges — and that might force a contradiction, or it might cause more vertices to become clothed.

Experiment with a modified step H11, which *maximizes* d in cases where $d < 2$ is impossible, by testing the modified algorithm on the benchmarks of Table 1.

130. [*20*] At the beginning of step H11, define the "current graph" G' to be the graph whose vertices are the currently visible vertices, and whose edges are (i) the edges of G that haven't been deleted, and (ii) the edges v — `MATE`(v) for all outer vertices v. Prove that we could safely jump to step H14 if G' isn't Hamiltonian.

Halin graph
preorder
regular q-gon
nonintersecting chords
chords
planar
isomorphic
uniformly Hamiltonian
Hamiltonian, uniformly
nonintersecting chords
pi, as random source
regular 100-gon
Halin graphs
Hamiltonian path
articulation point
planar
MRV heuristic
branching
benchmarks
current graph
visible vertices

132. [*15*] Why can't '`a`' appear in the name of a closed knight's tour's bunch?

133. [*M21*] Use "Burnside's Lemma" to determine the number of canonical bunches.

888. [*25*] Work in progress: To be an exercise about the giraffe census.

▶ **150.** [*22*] If G is a digraph on n vertices, define $\widehat{G}$ by adding two vertices s and t, with $2n$ additional arcs $s \longrightarrow v$ and $v \longrightarrow t$ for all v in G; also $t \longrightarrow s$.

a) Show that G has a Hamiltonian path if and only if $\widehat{G}$ has a Hamiltonian cycle.

b) Prove that if G has no cycles, it has at most one Hamiltonian path.

c) True or false: Algorithm B will handle $\widehat{G}$ in linear time when G is acyclic.

151. [*M25*] If G is a digraph with m arcs, n vertices, and p cycles, show that we need at most $O\bigl(n^p(m+n)\bigr)$ steps to test whether or not G has a Hamiltonian path.

198. [*20*] Given a bipartite graph G with n vertices in each part, construct an exact cover problem with $3n$ primary items u^-, u^+, v: two for each vertex u in the first part, and one for each vertex v in the second part. Let the options be '$u^-\ v\ w^+$', for all triples with $u \mathrel{-\!\!-} v \mathrel{-\!\!-} w$ and $u \neq w$.

a) What do the solutions to this exact cover problem represent?

b) Experiment with this construction when G is the 6 × 6 knight graph.

▶ **199.** [*27*] How many 8 × 8 closed knight's tours have the property that moves k and $32+k$ occupy the same column, for $1 \le k \le 32$? *Hint:* Define an exact cover problem.

200. [*24*] Find a knight's tour whose step matrix has

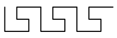
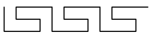
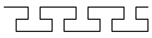
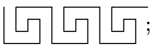
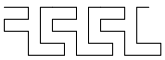
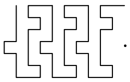
$$a_{11}=1,\ a_{16}=16,\ a_{32}=64,\ a_{52}=32,\ a_{71}=33,\ a_{83}=34,$$

and such that $1 \le a_{ij} \le 18$ implies $a_{(9-i)(9-j)} = 50 - a_{ij}$. (The latter condition means that moves 1, 2, ..., 18 are rotated 180° from moves 49, 48, ..., 32.)

▶ **201.** [*24*] (G. E. Carpenter, 1881.) Find a knight's tour for which the top row of the step matrix is '1 4 9 16 25 36 49 64'.

▶ **250.** [*M27*] Exactly how many Hamiltonian cycles are possible in the *Sierpiński gasket graph* $S_n^{(3)}$? (See Fig. 113, near 7.2.2.3–(69).) *Hint:* There is a fairly simple formula.

▶ **260.** [*25*] An $m \times n$ *meander frieze* is a Hamiltonian cycle of $P_m \mathbin{\square} C_n$ that isn't also a Hamiltonian cycle of $P_m \mathbin{\square} P_n$. For example, a few of the possibilities are


(i) 3 × 3
;

(ii) 3 × 4
;

(iii) 3 × 4
;

(iv) 4 × 4
;

(v) 5 × 4
;

(vi) 7 × 3
.

(Such friezes were often used as ornamentation on Greek vases of the "late Geometric" period, c. 750 B.C. For example, the famous Dipylon Amphora was decorated in part with the unusual frieze (vi) and several copies of (i) and (v).)

a) The *margins* of a meander frieze are the sequences $v_1 \ldots v_{m-1}$ and $h_1 \ldots h_n$, where there are v_i edges between rows $i-1$ and i and h_j edges between columns $j-1$ and j. For example, the margins of (iv) are $v_1v_2v_3 = 242$ and $h_1h_2h_3h_4 = 1331$. True or false: v_i is always even and h_j is always odd.

b) Prove that no $m \times n$ meander frieze has m even and n odd.

c) A meander frieze is *reduced* if it doesn't have $v_i = v_{i+1} = n$ or $h_j = h_{j+1} = m$ for any i or j. (For example, (ii) reduces to (i) because it has $h_2 = h_3 = 3$.) Find an example of an unreduced frieze for which $v_2 = v_3 = n$.

d) Draw all the meander friezes with $m \le 8$ and $n = 3$. Are any of them symmetric?

bunch
Burnside's Lemma
canonical bunches
Hamiltonian path
bipartite
exact cover problem
knight graph
step matrix
rotated 180°
near symmetry
Carpenter
Sierpiński gasket graph
meander frieze
frieze
Greek vases
vases
Geometric
Dipylon Amphora
margins
parity
reduced
symmetric

e) An $m \times n$ meander frieze is *periodic* if it's the same as d copies of an $m \times (n/d)$ meander frieze, for some proper divisor of n. Draw all of the nonperiodic, reduced meander friezes with $m = 3$ and $n \leq 8$. Which of them are symmetric?

f) How many automorphisms does the graph $P_m \,\square\, C_n$ have, when $m, n \geq 3$?

g) Count the nonisomorphic, nonperiodic, reduced meander friezes with $3 \leq m, n \leq 7$.

h) How many of the friezes in (g) are symmetrical?

i) Draw all of the friezes in (g) that have fourfold symmetry.

periodic
automorphisms
fourfold symmetry
torus

261. [*M21*] Describe the nonisomorphic Hamiltonian cycles of the torus $C_3 \,\square\, C_4$.

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

AUBREY
Kowalewski
threefold symmetries
complex plane
golden ratio ϕ
Gerbracht

> *After [this] way of Solving Questions, a man may steale a Nappe,*
> *and fall to worke again afresh where he left off.*
>
> — JOHN AUBREY, *An Idea of Education of Young Gentlemen* (c. 1684)

SECTION 7.2.2.4

1. Established conventions promote communication, so they outweigh convenience. [And we could save even more syllables by saying “HC” and “HP.” Many authors now save two syllables by saying just “Hamilton.”]

2. True (except in the trivial case where G has a single vertex). In fact, the number of Hamiltonian paths in G is the number of Hamiltonian cycles in G'; the number of Hamiltonian paths between u and $v \neq u$ in G is the number of Hamiltonian cycles in G' that include the edges by which u and v are joined to the new vertex.

3. The 12 vertices of Fig. 121(a) are named ij for $i \neq j$ and $1 \leq i, j \leq 4$. If $ij \longrightarrow i'j'$ then $ji \longrightarrow j'i'$ and $(i\alpha)(j\alpha) \longrightarrow (i'\alpha)(j'\alpha)$, where α is the permutation (123). We also have $12 \longrightarrow 23$, $12 \longrightarrow 34$, $12 \longrightarrow 41$, $12 \longrightarrow 42$, $14 \longrightarrow 42$. These rules define all edges.

The 20 vertices of Fig. 121(b) are named ij for $i \neq j$ and $1 \leq i, j \leq 5$. If $ij \longrightarrow i'j'$ then $ji \longrightarrow j'i'$ and $(i\sigma)(j\sigma) \longrightarrow (i'\sigma)(j'\sigma)$, where σ is the permutation (12345). We also have $12 \longrightarrow 35$, $12 \longrightarrow 43$, and $13 \longrightarrow 24$. These rules define all of the edges.

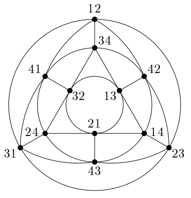
[See A. Kowalewski, *Sitz. Akad. Wiss. Wien* (IIa), **126** (1917), 67–90, 963–1007. See exercise 7.2.2.1–136 for a 3D-geometry-based representation scheme.]

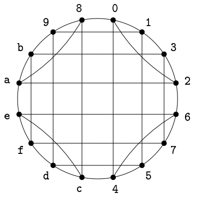
4. It remains unchanged after 180° rotation about any of the following lines: (i) from $\frac{13+45}{2}$ to $\frac{31+54}{2}$; (ii) from $\frac{14+53}{2}$ to $\frac{41+35}{2}$; (iii) from $\frac{15+34}{2}$ to $\frac{51+43}{2}$.

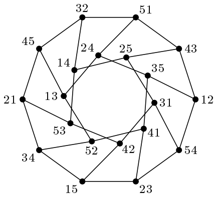
There also are remarkable threefold symmetries of a different kind: Color the edges of the cycle alternately red and green; color the other edges blue. Then a 120° rotation about any of the lines from 21 to 12, 23 to 32, 24 to 42, or 25 to 52 will permute the colors cyclically(!). That will yield green-blue and blue-red cycles (see exercise 24).

5. We can redraw the edges $\{12 \longrightarrow 35,\ 51 \longrightarrow 24,\ 45 \longrightarrow 13,\ 21 \longrightarrow 34,\ 53 \longrightarrow 42\}$ so that they lie outside the circle. [A cubic planar graph with a Hamiltonian cycle can always be drawn as a circle, with some of the unused $n/2$ noncrossing edges inside and the others entirely outside. See exercise 103 for more about planarity.]

6, 7, 8.







To determine the distances and angles needed for the third drawing above, assuming that vertex ‘12’ is point 1 in the complex plane and that ‘35’ is point $re^{i\theta}$, solve $|1 - re^{i\theta}|^2 = |1 - e^{i\pi/5}|^2 = |re^{i\theta} - re^{i(\theta-2\pi/5)}|^2$. The answer (see exercise 1.2.8–19) is $r = 5^{-1/4}\phi^{-1/2} \approx .5257$, $\theta = \frac{\pi}{4} - \frac{1}{2}\arctan\frac{1}{2} \approx .5536$. [E. H.-A. Gerbracht, *Kolloquium über Kombinatorik*, Universität Magdeburg (15 November 2008).]

9. (a) We can assume by symmetry that the cycle begins at vertex 12, having just come from 35. Case (i) takes us to 54, 23, 41, 35; oops! In case (ii) it’s 54, 31, 25, 14; now 43 is stranded. In case (iii) the moves to 54, 31, 42, 53, 21, 45, 13 force the cyclic path $51 \longrightarrow 43 \longrightarrow 25 \longrightarrow 14 \longrightarrow 32 \longrightarrow 51$. The opening moves 54, 23, 15, 34 in cases (iv)–(vi) force the ending to be . . . , 51, 24, 13, 52, 41, 35; so those cases are ruled out.

(b) The only remaining possibilities are $(\mathrm{LLLRRRLRLR})^2$ and $(\mathrm{RRRLLLRLRL})^2$.

10. All but 23, 24, 25, 31, 41, and 51. (There are 20 Hamiltonian paths from 12 to 35, in spite of the "uniqueness" of exercise 9. There are only six such paths from 12 to 21.)

11. Let $a_j = (2j)'$, $b_j = 2j$, $c_j = 2j+1$, and $d_j = (2j+1)'$. Notice that $\mathrm{GP}(2q,2)$ is a graph for $q \ge 3$, a multigraph for $q < 3$. The ungeneralized Petersen graph is $\mathrm{GP}(5,2)$.

A Hamiltonian path P can be characterized by its endpoints and its 3-bit "states"

$$s_j = [a_j \mathrel{-\!\!-} a_{j'} \in P][c_j \mathrel{-\!\!-} b_{j'} \in P][d_j \mathrel{-\!\!-} d_{j'} \in P], \quad 0 \le j < q, \quad j' = (j+1) \bmod q.$$

For example, with endpoints $\{a_0, a_1\}$ and $q = 3$, the states $(s_0, s_1, s_2) = (011, 111, 010)$ can arise only from the path $a_0 \mathrel{-\!\!-} b_0 \mathrel{-\!\!-} c_2 \mathrel{-\!\!-} d_2 \mathrel{-\!\!-} d_1 \mathrel{-\!\!-} d_0 \mathrel{-\!\!-} c_0 \mathrel{-\!\!-} b_1 \mathrel{-\!\!-} c_1 \mathrel{-\!\!-} b_2 \mathrel{-\!\!-} a_2 \mathrel{-\!\!-} a_1$. Moreover, the states $(s_0, s_1, \ldots, s_{q-2}, s_{q-1}) = (011, 111, \ldots, 111, 010)$ yield a path from a_0 to a_1 *whenever* $q \ge 3$. (Adding the edge $a_1 \mathrel{-\!\!-} a_0$ then gives a Hamiltonian cycle.) Those same states also define a Hamiltonian path from a_0 to c_1.

Only certain state transitions $s_j \to s_{j'}$ are possible. For example, parity is preserved if $\{a_{j'}, b_{j'}, c_{j'}, d_{j'}\}$ contains no endpoint; and the only such legal transitions are

$$001 \to 100,\ 001 \to 111,\ 010 \to 111,\ 100 \to 001,\ 111 \to 010,\ 111 \to 100,\ 111 \to 111;$$
$$000 \to 101,\ 011 \to 110,\ 101 \to 000,\ 101 \to 011,\ 110 \to 011,\ 110 \to 101.$$

Certain additional restrictions also apply. For example, $110 \to 011$ can be used at most once, or it will "disconnect" the path. The sequence $001 \to 111 \to 010$ forces a 5-cycle.

Parity is preserved also when *both* endpoints lie in $\{a_{j'}, b_{j'}, c_{j'}, d_{j'}\}$. For example, we get a path from a_0 to d_0 for all $q \ge 2$ from the sequence $(111, \ldots, 111, 010)$.

Transition rules at parity changes are also easy to work out. For example, if $a_{j'}$ is an endpoint the legal transitions are

$$000 \to 001,\ 011 \to 010,\ 011 \to 100,\ 011 \to 111,\ 110 \to 001;$$
$$001 \to 000,\ 001 \to 011,\ 010 \to 011,\ 010 \to 101,\ 111 \to 000,\ 111 \to 011.$$

It turns out that Hamiltonian paths from a_0 to v exist except when v lies in B_q, where $B_q = \{a_j \mid j \bmod 3 = 0,\ 0 < j < q\}$ when $q \bmod 3 = 0$; $B_q = \{a_j \mid j \bmod 3 = 2,\ 0 < j < q\}$ when $q \bmod 3 = 1$; and $B_q = \{a_j \mid j \bmod 3 \ne 1,\ 0 < j < q\} \cup \{c_0, c_{q-1}\} \cup \{b_j \mid j \bmod 3 = 1,\ 0 < j < q\}$ when $q \bmod 3 = 2$. (Unless $q < 4$: $B_3 = \{b_1, b_2\}$.)

12. Consider the state transitions in answer 11. The legal cycles of odd-parity states are of two kinds, namely $(010, 111^*, 111)^k$ and $(001, 111^*, 100)^k$; here '111^*' stands for zero or more repetitions of 111. Two legal cycles of even-parity states exist when $q = 3k+2$, namely $(000, 101, (011, 110, 101)^k)$ and $(110, (011, 110, 101)^k, 011)$.

The number of Hamiltonian cycles is the number of legal cycles of states, and we can enumerate them by using generating functions. The answer turns out to be $2L_q - 2 + 2q[q \bmod 3 = 2]$, where $L_q = F_{q+1} + F_{q-1}$ is the qth Lucas number.

14. (a) In fact, let H be *any* induced subgraph whose vertices all have degree 3 except for exactly two vertices of degree 2. The other edges from those two must be true in every win, or we'd have a cycle entirely within H.

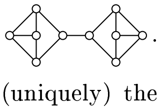
(b) The connecting edges are 1, and so are many of the internal edges. Thus internal cycles will appear when $x+y+z$ is 0, 2, or 3. (But if $x+y+z = 1$ we easily have a path through all the internal vertices.)

(c) The long horizontal edges must be true, because consecutive true vertical edges would yield a short cycle. Hence the edge values at the left and right are x, $\bar{x}$, x, $\bar{x}$, x and y, $\bar{y}$, y, $\bar{y}$, y. If $x = y$ we'd have a 4-cycle or two 8-cycles.

multigraph
parity
generating functions
Lucas number
Fibonacci numbers

(d) Hook C_m to C_1. Then insert XOR gadgets to ensure that all appearances of the same variable have consistent values. [This construction can be extended so that G is not only cubic but planar, and triconnected, with at least five sides on every face. See M. R. Garey, D. S. Johnson, and R. E. Tarjan, *SICOMP* **5** (1976), 704–714.]

16. (1, 2, 5, 19) connected cubic graphs on (4, 6, 8, 10) vertices are essentially distinct (not isomorphic); we'll study how to generate them in Section 7.2.3. They all are Hamiltonian except for two of order 10. One of the latter is the famous Petersen graph (Fig. 2(e) near the beginning of Chapter 7), which also is nonplanar.

The other "smallest" non-Hamiltonian example is actually planar: . [Arun Giridhar verified in 2015 that a 16-vertex variant of this graph, consisting of three 5-vertex diamonds joined to a central vertex, is (uniquely) the smallest cubic graph that has no Hamiltonian *path*.]

18. False. For example, consider $0 — 1 — 2 — 3 — 4 — 5 — 0$, $0 — 2 — 4 — 0$.

20. (a) The condition holds when a_k is the number of k-sided faces *inside* the n-cycle. For it's certainly true when there's just one such face ($a_k = [k = n]$). And if a new chord is added to the graph, breaking an inner p-face into a q-face and an r-face where $q + r = p + 2$, $\sum_k (k-2)a_k$ changes by $(q-2) + (r-2) - (p-2) = 0$.

[The number $a'_k = \alpha_k - a_k$ of k-faces *outside* the cycle is also a solution to Kirkman's conditions. Indeed, we always have $\frac{1}{2}\sum_k (k-2)\alpha_k = \frac{1}{2}\sum_k k\alpha_k - \sum_k \alpha_k = \langle\text{edges}\rangle - \langle\text{faces}\rangle = \langle\text{vertices}\rangle - 2$ in a connected planar graph.]

(b) We can assume that the missing vertex is outside the cycle; and $3a_5$ can't equal $19 - 2$. (The dodecahedron does have cycles of lengths $\{5, 8, 9, 10, \ldots, 17, 18\}$.)

(c) We can assume that neither cycle is inside the other. A cycle that contains exactly a pentagons has length $3a + 2$; and $(3a + 2) + (3b + 2)$ can't equal 20.

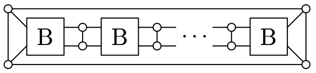
(d) Let G' be G without the edge $a — b$, and with two additional vertices of degree 2: one inserted between a and f, another between b and c. Any Hamiltonian cycle in G that omits $a — b$ corresponds to a Hamiltonian cycle in G'. But G' isn't Hamiltonian, because $\alpha'_k = [k = 4] + 7[k = 5] + [k = 11]$ and $2a'_4 + 3a'_5 + 9a'_{11} \neq 18 - 2$.

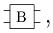
(e) Graph (i) has $\alpha_k = [k = 4] + 20[k = 5] + 2[k = 11]$; graph (ii) has $\alpha_k = [k = 4] + 18[k = 5] + 4[k = 8]$. So both of them fail Kirkman's test (a). Graph (iii), with $\alpha_k = 18[k = 5] + 3[k = 6] + 3[k = 8]$, passes the test only if $a_6 = 0$ or 3. But the three 6-edged faces can't all be inside or outside the cycle, since they share a common point.

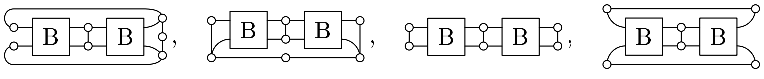
[*Historical notes:* As noted near the beginning of this chapter, Kirkman actually studied full-length cycles in convex polyhedra before Hamilton began to toy with such ideas. [See *Philosophical Transactions* **146** (1856), 413–418.] Graphs (ii) and (iii) in part (e) are due to É. Ya. Grinberg, *Latviĭskiĭ Matematicheskiĭ Ezhegodnik* **4** (1968), 51–58, who rediscovered Kirkman's long-forgotten criterion. Graph (i) was found as part of an exhaustive computer search by G. B. Faulkner and D. H. Younger, *Discrete Math.* **7** (1974), 67–74, who also established that Grinberg's (iii) is the unique smallest cubic planar graph that is *cyclically 5-connected*: It cannot be broken into two components each containing a cycle unless at least five edges are removed. (Graph (ii) clearly has four automorphisms; and graph (iii), obtained by adding a single edge, actually has six, although that isn't obvious from the diagram. If we add *another* edge at the right, in the mirror-image position, we get a 46-vertex graph with 36 Hamiltonian cycles. Of course each of those cycles uses both of the edges that were added to (ii).)]

planar
triconnected
Garey
Johnson
Tarjan
isomorphic
Petersen graph
Giridhar
Hamiltonian *path*
Historical notes
Hamilton
Grinberg
Faulkner
Younger
cyclically 5-connected
automorphisms

21. Among many possibilities, the simplest are perhaps the $(20n+2)$-vertex graphs

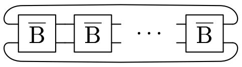
$\mathrm{FY}_n =$ 

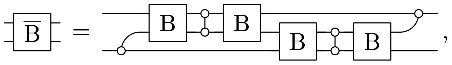
made from n copies of an 18-vertex gadget , where graph (i) in exercise 20(e) is FY_2 and is illustrated there. In general, FY_n has $\alpha_k = [k=4] + 10n[k=5] + 2[k=5n+1]$; so it fails Kirkman's test whenever $n \bmod 3 = 2$. Further analysis, based on the fact that each of the four graphs



also fails Kirkman's test, shows that FY_n is actually non-Hamiltonian for *all* $n > 1$.

(Faulkner and Younger went on to show that the $78m$-vertex graphs



where 

are not only non-Hamiltonian, they can't be covered by fewer than m disjoint cycles.)

Grinberg's paper of 1968 described non-Hamiltonian cubic planar graphs having $(14\cdot 4^s - 10\cdot 3^s)(3t-1)$ vertices, for any $s,t>0$. His graphs are noticeably harder than FY_n for a computer to analyze; even the case $(s,t)=(2,1)$ is quite a challenge.

24. (a) $K_4 \oplus K_4$ is disconnected (and K_4 is perfectly Hamiltonian). The others are at least Hamiltonian, and we can number the vertices so that $0 — 1 — \cdots — 7 — 0$. There are five nonisomorphic possibilities: *Case 1,* $0 — 2$, $1 — 3$, $4 — 6$, $5 — 7$*:* 16 auts, 4H. [Translation: 16 automorphisms and 4 Hamiltonian cycles.] *Case 2,* $0 — 2$, $1 — 5$, $3 — 6$, $4 — 7$*:* 12 auts, 6H, perfect (two sets of three). *Case 3,* $0 — 2$, $1 — 5$, $3 — 7$, $4 — 6$*:* 4 auts, 3H, planar, perfect. *Case 4,* $0 — 3$, $1 — 6$, $2 — 5$, $4 — 7$*:* 48 auts, 6H, planar [the 3-cube]. *Case 5,* $0 — 4$, $1 — 5$, $2 — 6$, $3 — 7$*:* 16 auts, 5H.

(b) Let a_k be the number of k-faces inside none of the three Hamiltonian cycles; let b_k, c_k, d_k be the number that are inside cycles $\{1,2\}$, $\{1,3\}$, $\{2,3\}$, respectively. Then Kirkman's criterion for cycle 1 is satisfied by b_k+c_k and a_k+d_k, the number of faces respectively inside or outside. Similarly, it's satisfied for cycle 2 by b_k+d_k and a_k+c_k; for cycle 3 by c_k+d_k and a_k+b_k. Let $A=\sum_k(k-2)a_k$, …, $D=\sum_k(k-2)d_k$; we've shown that $A{+}B = A{+}C = A{+}D = B{+}C = n{-}2$. [See Grinberg's paper in answer 20.]

[Similarly, an r-regular graph is perfectly Hamiltonian if its edges can be r-colored in a such a way that all $\binom{r}{2}$ pairs of colors yield Hamiltonian cycles. (The 4-regular 6-vertex "Star of David" graph $L(K_4)$ is a good example; so is the 5-regular K_6.) Such graphs are also called P1F, "perfectly 1-factorable," because two 1-factors — also known as perfect matchings — are called perfect if they yield a Hamiltonian cycle, and because an r-regular graph with $\chi(L(G)) = r$ is called 1-factorable. The pioneering explorations of A. Kotzig (see *Theory of Graphs and its Applications*, ed. by M. Fiedler (1964), 63–82) have led to a large literature with many provocative problems still unsolved (see A. Kotzig and J. Labelle, *Annales des Sciences Math. du Québec* **3** (1979), 95–106); for an excellent survey see A. Rosa, *Mathematica Slovaca* **69** (2019), 479–496. Answer 124 below, Case 2, proves incidentally that cubic Halin graphs are perfectly Hamiltonian.]

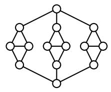
27. (a) This follows directly from exercise 20(d). (In general, we get a "forcing" gadget from *any* graph that has a "forced" edge, by removing any vertex of that edge.)

(b) Insert a degree-2 vertex into each of those spokes and apply 20(a).

(c) Two nonconsecutive spokes are forced to be in any Hamiltonian path.

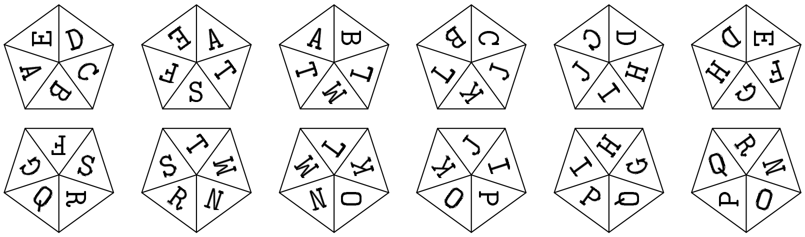
gadget
Kirkman
Faulkner
Younger
Grinberg
3-cube
Grinberg
Historical notes
r-regular graph
"Star of David" graph
P1F
1-factors
perfect matchings
Kotzig
Fiedler
Labelle
strongly H graphs, see perfectly H graphs
Rosa

(d) No. One of the five 4-faced regions touches the unique 9-faced region. Its other neighbors have respectively $\{5,8,8\}$, $\{5,7,7\}$, $\{5,7,8\}$, $\{7,8,8\}$, $\{7,7,7\}$, $\{7,7,8\}$ faces.



Historical notes: A cubic graph that can be disconnected by removing one edge is clearly non-Hamiltonian. Many cyclically *2-connected* planar cubic graphs, such as the example shown, also have no Hamiltonian cycle. However, P. G. Tait investigated numerous cyclically *3-connected* planar cubic graphs — the "true" polyhedra — and found Hamiltonian cycles easily. So he conjectured that such cycles always exist, and he pointed out that the famous "Four Color Theorem" would then follow. (See §15 and §16 of his paper cited in answer 35.) Tait's conjecture was believed for many years, until W. T. Tutte [*J. London Math. Soc.* (2) **21** (1946), 98–101] found a 46-vertex counterexample by putting together three Tutte gadgets. The smaller graphs in (c) were found independently in 1964 by D. Barnette, J. Bosák, and J. Lederberg; those graphs are the *only* counterexamples with fewer than 40 vertices [see D. A. Holton and B. D. McKay, *J. Combinatorial Theory* **B45** (1988), 305–319]. Every counterexample has a face with more than 6 vertices [F. Kardoš, *SIAM J. Discrete Math.* **34** (2020), 62–100]; in particular, all "fullerene graphs" are Hamiltonian.

30. We can use the Ls and Rs of answer 8 as a guide:



(The author cherishes a 3D-printed object like this, received as a surprise gift in 2016.)

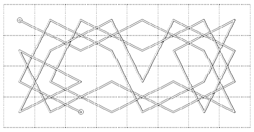
33. $nH+h$ — one for every Hamiltonian path in G (cyclic or not). (Thus an algorithm that finds all Hamiltonian cycles can readily be adapted to find all Hamiltonian paths.)

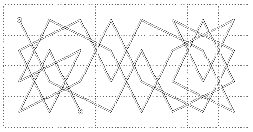
35. (a) The tour lines divide the plane into regions. Every such region can be assigned a rank, representing its distance from the outside. (More precisely, the rank is the minimum number of tour lines crossed by any path in the plane that goes from a point in the region to a point outside the chessboard, without passing through any of the tour's intersection points.) Then, as you walk along the tour, make your thread go on top at an intersection if and only if the region on your left has odd rank. [See P. G. Tait, *London, Edinburgh, and Dublin Philosophical Magazine* **17** (1884), 30–46, §19.]

(b) One endpoint is in the outside region, but the other is in a region of rank 4. Any artificial path that connects the two, and crosses k tour lines, will lead to a drawing with k exceptions when the artificial path is removed. Conversely, the exceptional tour segments in any drawing can be crossed by an artificial connection path together with zero or more artificial cycles; so there must be at least 4 exceptions.

(There's no problem in (2), because each endpoint lies in the outer region.)

36. In (i), $\sigma_{22} \leftrightarrow \sigma_{24}$ and $\sigma_{25} \leftrightarrow \sigma_{27}$. In (ii), 'lots' matches 'lost':

(i)  ;

(ii)  .

Historical notes
isthmus
bridge
cyclically *2-connected*
bridge, wheatstone
Tait
cyclically *3-connected*
Four Color Theorem
Tutte
Barnette
Bosák
Lederberg
Holton
McKay
Kardoš
fullerene graphs
author
3D-printed
all Hamiltonian paths
rank
Tait

37. Starting at cells (1, 2, 3, 4) of row 1 we obtain respectively (7630, 2740, 2066, 3108) tours. Starting at cells (1, 2, 3, 4) of row 2 we obtain none. Thus there are exactly $4 \cdot (7630 + 2740 + 2066 + 3108) = 62{,}176$ tours, all of which are open because they begin and end in the top or bottom row. (Among all such tours, 1904 cannot be represented by a single Rudraṭa-style sloka because all 32 syllables of such a sloka would have to be identical! Only the example in answer 36(ii), and its reversal after 180° rotation, are representable by a sloka that has 12 distinct syllables.)

open
Rudraṭa
poetic license
Murray
silver general
shōgi
Japanese chess

38. One knight jumps like three rookwise steps.
Past sore too mean; so, just for free,
Hops here, turns there, flies each goose now.
Can't place last word? Won't find the sea.
One, two, three, four! See each word here:
Jumps so wise now find their place passed.
Terns can't soar, like flies the free rook;
Goose steps just won't mean knight hops last.

40. Meet me, you fool; trip some word up;
Eat, see if autumn is a mess.
To forgo this ordeal, I cheat:
Won three games like dice, card-trick, chess.

One, two, three, four! Games go like this:
Dice or card deal, tricky chess cheat,
Mess up; a word is some dumb trip.
Awful if you see me eat meat.

[Sloka 16 in Rudraṭa's *Kāvyālaṅkāra* can be interpreted as two poorly joined quarter-tours of an elephant. See Murray's *History of Chess*, pages 54 and 55.]

(A "silver general" in shōgi (Japanese chess) has the same moves as an elephant.)

41. (a) There are $(m-1)n$ "trunk" arcs from (i,j) to $(i-1,j)$, plus $4(m-1)(n-1)$ "leg" arcs from (i,j) to $(i \pm 1, j \pm 1)$; total $(m-1)(5n-4)$.

(b) Yes: If and only if $m = 2$ and n is even. (Use just two trunk moves.)

(c) In fact, the solution in Fig. A–18(a) is the *only* such Hamiltonian path on E_{48}.

(d) If not, every vertex of row 1 is of type A (↙↑) or B (↑↘) or C (↙↖) or D (↗↘) within the path. There's a B at the left and an A at the right. The adjacent pairs AD, BD, CA, CB are not permitted, nor are the near-adjacent pairs B∘A, B∘C, C∘D, D∘A, D∘C (with one vertex intervening). Furthermore the substrings $\mathrm{B(CD)^*A} = \{\mathrm{BA}, \mathrm{BCDA}, \mathrm{BCDCDA}, \ldots\}$ are forbidden, because these are closed cycles and $m > 2$.

But no string of A's, B's, C's, and D's obeys all of those restrictions.

(e) The same proof works, with types A (↘↕), B (↕↙), C (↘↗), and D (↖↙).

(f) The vertices of (d) are joined by one of type E (↙) or F (↘). The leftmost is either B or F; the rightmost is either A or E. Cases B∘E, D∘E, F∘A, F∘C are excluded, in addition to those of (d). Exactly $n\,[n\,\mathrm{even}]$ such sequences are possible, having the forms $\mathrm{F(CD)^*A}$, $\mathrm{B(CD)^*ED(CD)^*A}$, $\mathrm{B(CD)^*CF(CD)^*A}$, or $\mathrm{B(CD)^*E}$.

Each of these has exactly one unsaturated vertex in row 2. Thus there are one or two possible moves to row 3, and we've effectively reduced m to $m-2$.

(g) Now the vertices of (e) are joined by one of type E (↖) or F (↗) or G (↕). The leftmost is either B, F, or G; the rightmost is A, E, or G. The new forbidden substrings are AE, BE, CG, FA, FB, GD, C∘E, F∘D. Six species of solutions exist, namely $\mathrm{GA^*(CD)^*A}$, $\mathrm{B(CD)^*B^*G}$, $\mathrm{B^*FD(CD)^*A}$, $\mathrm{BCD(CD)^*B^*FD(CD)^*A}$, $\mathrm{B(CD)^*CEA^*}$, and $\mathrm{B(CD)^*CEA^*(CD)^*CDA}$. The solutions containing $\mathrm{A^*}$ or $\mathrm{B^*}$ work when n is odd.

directed graphs
ZDD
oriented cycles
generating function
Rudrața
open tour
closed tours

Again we reduce m to $m-2$ and continue. (By induction, m must be even.)

(h) Let $A^{(m)}$ be the $n \times n$ matrix where $a_{ij}^{(m)}$ is the number of Hamiltonian paths from $(1,i)$ to (m,j); let $B^{(m)}$ be analogous, where $b_{ij}^{(m)}$ counts paths from (m,i) to $(1,j)$. (These matrices are symmetric about both diagonals, because the left-right and top-bottom reflections of any elephant path are elephant paths, possibly reversed.) We have

$$a_{ij}^{(2)} = \bigl([i=j=1] + [i=j=n] + [|i-j|=1 \text{ and } \max(i,j) \text{ odd}]\bigr)[n \text{ even}],$$
$$b_{ij}^{(2)} = \begin{cases} [i \text{ odd}][j \text{ even}][i<j] + [j \text{ odd}][i \text{ even}][j<i], & \text{if } n \text{ is even,} \\ [i \text{ odd}][j \text{ odd}] \max\bigl([i=1],[i=n],[i\neq j]\bigr), & \text{if } n \text{ is odd,} \end{cases}$$

by (f) and (g). Moreover, by considering moves between two-row subgraphs,

$$A^{(m+m')} = A^{(m)} X A^{(m')} \quad\text{and}\quad B^{(m+m')} = B^{(m)}(X+I)B^{(m')}, \quad\text{where } x_{ij} = [|i-j|=1].$$

For example, there are $\sum A^{(4)} + \sum B^{(4)} = 14 + 120 = 134$ tours on a 4×8 board.

42. The technology of exercise 7.1.4–226 can be extended to directed graphs in a straightforward way. It constructs a ZDD of about 1.3 meganodes for all oriented cycles in the 8×8 elephant digraph, and shows that there are exactly 277,906,978,347,470 of them. The generating function by cycle length is $98z^2 + 205z^4 + 698z^6 + 3853z^8 + \cdots + 50128559z^{60} + 6544z^{62}$. If we say that trunk moves have weight 2 while other moves have weight 3, we find (by computing maximum-weight cycles) that exactly four of the 6544 62-cycles have only eight trunk moves. All four of these solutions are equivalent under reflection to Fig. A–18(b).

(Figure A–18(c) shows an interesting symmetrical 60-cycle that omits the corners and has just *four* trunk moves.)

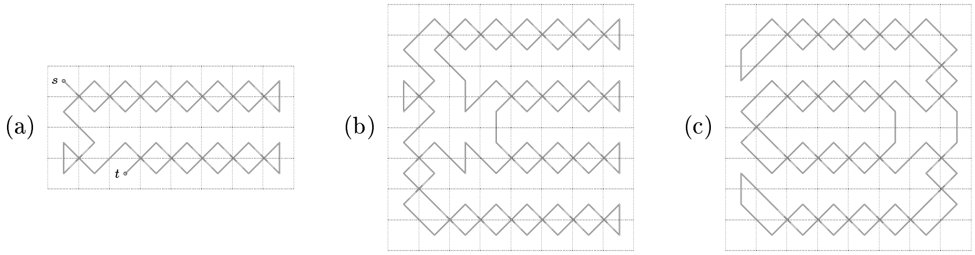


Fig. A–18. Noteworthy paths and cycles for elephants.

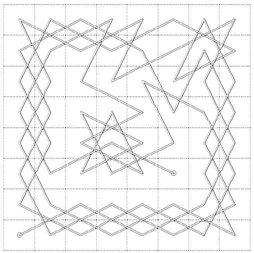
44. We can use Rudrața's half-tour twice:

Bah dee boo dai hao fuh hoe fay, bee doo bai fao huh foe hay dah?
  Fee hah day boe foo hai dao buh, fai bao duh hoo doe bay fah hee.
Lah mee loo mai sao nuh soe nay, lee moo lai nao suh noe say mah?
  Nee sah may loe noo sai mao luh, nai lao muh soo moe lay nah see!

(This is an open tour. See exercise 199 for *closed* tours that rhyme just as perfectly.)

Ibn Manī‘
Murthy
Rudraṭa
Ratnākara
Deśika
de Jaenisch
von Warnsdorf
backtrack algorithm

46. This tour is rather like that of Ibn Manī‘ in (1):



[G. S. S. Murthy, in *Resonance* **25** (2020), 1095–1116, comments further on the work of Someshvara, and gives excellent translations of the Sanskrit verses by Rudraṭa, Ratnākara, and Vedānta Deśika.]

50. In fact, the text's "merry chase" need not end at cell 22; by swapping **23** ↔ **25** we get a tour that ends at 02. The other seven choices of **9** and **10** can lead, similarly, to either 22 or one of $\{02, 11, 13, 20, 24, 31, 33\}$. Each case completes an open tour.

[See page 280 of de Jaenisch's book, for his analysis of 5 × 5 paths.]

51. Again $v_1v_2v_3v_4 = 00\ 12\ 04\ 23$ without loss of generality. Now $t_1 = 44$ forces $v_5 = 31$, and there's a tie for v_6. If $v_6 = 43$, the path continues $v_7 \ldots v_{15} = 24\ 03\ 11\ 30\ 42\ 34\ 13\ 01\ 20$; and v_{16} is 32 or 41. The former case forces $v_{17} = 40$, hence "shutting out" 44; it leads to four paths, each ending at v_{24}. But the latter case leads to three paths, two of which end with $v_{25} = 44$ (yea) and one that ends with $v_{21} = 44$ (boo).

On the other hand if $v_6 = 10$ we get $v_7 \ldots v_{13} = 02\ 14\ 33\ 41\ 20\ 01\ 13$ and then $v_{14}v_{15}v_{16} = 21\ 40\ 32$ or $32\ 40\ 21$. Either case shuts 44 out. Ten continuations are possible, each of which involves 24 vertices — all but cell 44 (close but no cigar).

(The randomized algorithm of exercise 53 will yield a Hamiltonian path with probability $\frac{3}{16}$. If we set $\{t_1, t_2, t_3\} = \{23, 32, 44\}$ and $r = 3$, this probability rises to $\frac{3}{8}$.)

52. The algorithm acts just as if a double-target vertex t has been entirely removed from the graph, because `DEG(`t`)` will never be less than $2n$ in step W5.

53. W5′. [Is `DEG(`u`)` smallest?] If $t < \theta$, set $\theta \leftarrow t$, $v \leftarrow u$, $q \leftarrow 1$. Otherwise, if $t = \theta$, set $q \leftarrow q + 1$, then set $v \leftarrow u$ with probability $1/q$.

55. Ibn Manī‘ broke Warnsdorf's rule first when choosing $v_{14} = 41$ instead of 06. His choices for v_{26}, v_{27}, v_{38}, v_{39}, v_{45}, v_{48}, and v_{54} also broke the rule. But altogether, his "hug the edge" strategy followed it $\frac{55}{63} \approx 87\%$ of the time, so he probably had some of the same intuition that von Warnsdorf acquired later. Similarly, Someshvara deviated only eight times. But al-‘Adlī broke the rule 15 times, clearly thinking other thoughts.

56. If there's no remaining exit from u, there's no remaining entrance to u. Therefore Algorithm W will not find a Hamiltonian path unless it moves to u. We might as well do that, if our goal is simply to find a Hamiltonian path. But maybe we really want to find as long a path as possible, via Warnsdorf-like rules; then we can do better.

Vertex u is a dead end if and only if `DEG(`u`)` is 0 or n. Suppose v_k has just two neighbors, u and u', where `DEG(`u`)` $= 0$ and `DEG(`u'`)` $= n$. We presumably should choose $v_{k+1} = u'$ in such a case, because u' is one of the designated targets.

Thus we can improve the algorithm by changing '$t < \theta$' in step W5 to '$f(t) < f(\theta)$', where $f(0) = 2n + 1$, $f(n) = 2n$, $f(t) = t + 2$ for $t \geq 2n$, and $f(t) = t$ otherwise.

57. This is a simple backtrack algorithm, following the outline of Algorithm 7.2.2B. Leaves of the search tree correspond to the paths of Algorithm W. The probability of each node is the probability of its parent, divided by the family size.

59. Build 64 anti-Warnsdorf trees, as in exercise 57. There are 8 paths $v_1v_2\ldots v_{20}$ of length 19 (equivalent to each other via rotation/reflection), each occurring with probability $2^{-9}3^{-6} \approx .0000027$; they actually turn out to be cycles of length 20. At the other extreme, there are 24 anti-Warnsdorf paths as long as 53, of which 8 have probability $p = 2^{-20}3^{-5} \approx .0000000039$ and 16 have probability $3p$; representatives are shown below. The mean path length is ≈ 34.8, and the standard deviation is ≈ 5.3.

By contrast, Algorithm W will find a path that's longer than *every* anti-Warnsdorf path, more than 99.8% of the time. Its worst case has length 39; there are eight such paths, equivalent to the one shown, each obtained with probability 2^{-21}. (And if we use the modification of answer 56, the worst case length rises to 46; there are 40 such paths, of which the one shown is the most likely — its probability is $2^{-18}3^{-4}$. The total probability of all 40 cases for length 46 is $169 \cdot 2^{-23}3^{-3} \approx .00000075$.)

	20						
		18	1				
19		11	8	5	2		
	17	14	3	10	7		
	12	9	6	15	4		
		16	13				

	32	1	54	29		43	
	53	30	33	44	41	28	
31	34	15	2	7	10	45	42
52	21	8	5	12	3	40	27
35	16	23	14	9	6	11	46
22	51	20	17	4	13	26	39
	36	49	24	19	38	47	
50		18	37	48	25		

	32	1	42	29		53	
	43	30	33	52	41	28	
31	34	15	2	7	10	51	54
44	21	8	5	12	3	40	27
35	16	23	14	9	6	11	50
22	45	20	17	4	13	26	39
	36	47	24	19	38	49	
46		18	37	48	25		

		45	32	49			
46	31	48	1	20	33	50	37
17	44	19	10	5	36	21	34
30	47	16	7	2	9	38	51
43	18	27	4	11	6	35	22
26	29	12	15	8	3	52	39
	42	25	28	13	40	23	54
		14	41	24	53		

2	19	4	21			14	
5	22	1		15			
18	3	20					13
23	6		16				
32	17	30				12	
7	24	33		29			38
34	31	26	9	36	39	28	11
25	8	35	40	27	10	37	

3	36	5				23	
6	43	2		24			
37	4	35					22
44	7	42	1		25		
41	38	45	34			21	26
8	31	10	39	46	29	18	15
11	40	33	30	13	16	27	20
32	9	12	47	28	19	14	17

60. A tree constructed as in exercise 57 has two different forms, depending on whether s is a variable of type $\{a, d\}$ or $\{b, c\}$, because R_6 doesn't have as much symmetry as R_5.

(a) $\approx (.86, .74, .65, .57, .50)$ when $s = a_0$; $\approx (.90, .81, .71, .63, .56)$ when $s = b_0$. [For $q = 100$ these probabilities drop to about .000044 and .000050.]

(b) $\approx (.06, .05, .05, .04, .04)$ when $s = a_0$; $\approx (.07, .05, .05, .04, .04)$ when $s = b_0$.

(c) $\approx \bigl((.33, .29, .29, .24, .22), (.26, .26, .24, .19, .17)\bigr)$ when $s = a_0$, $t_1 = (b_0, a_1)$; and $\approx \bigl((.27, .23, .23, .18, .17), (.28, .21, .18, .15, .13)\bigr)$ when $s = b_0$, $t_1 = (a_0, c_0)$.

62. There are two choices for v_2; and the next moves must hug the edge. Thus the graph reduces either to $P_{m-1} \mathbin{\square} P_n$ or $P_m \mathbin{\square} P_{n-1}$, and we succeed by induction.

[It's necessary to require $r = 0$; for example, the algorithm can fail when $m = n = 4$, $s = (0, 0)$, $t_1 = (0, 3)$. The start vertex must also be a corner; consider $m = 4$, $n = 5$, $s = (0, 2)$. A similar proof shows that Algorithm W never fails on the convex triangular grid graphs produced by SGB's generator *simplex*$(n, n_1, n_2, n_3, 0, 0, 0)$, when starting at a corner. It also succeeds on the *dual* of that graph, provided that $n_1, n_2, n_3 < n$ and that a suitable starting vertex is specified; this is the graph whose vertices are the triangular faces inside the polygon, all of which have degree 2 or 3.]

63. Notice first that the standard Gray binary code $g(0)$, $g(1)$, …, $g(2^n - 1)$ is one of the paths that satisfy Warnsdorf's rule. For example, when $n = 3$ and $s = 000$, this path is 000, 001, 011, 010, 110, 111, 101, 100; and it has the delta sequence $\delta_0 \ldots \delta_6 =$ 0102010 of the ruler function ρ (see 7.2.1.1–(24)). Warnsdorf's rule for the 3-cube allows any $\delta_0 \in \{0, 1, 2\}$; if $\delta_0 = 0$, it allows $\delta_1 \in \{1, 2\}$; if $\delta_0\delta_1 = 01$, it forces $\delta_2\delta_3 = 02$; if $\delta_0\delta_1\delta_2\delta_3 = 0102$, it allows $\delta_4 \in \{0, 1\}$; and if $\delta_0 \ldots \delta_4 = 01020$, it forces $\delta_5\delta_6 = 10$.

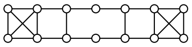
In general one can prove that if $\delta_{k-1} = \rho(k)$ for $1 \le k < r$, then Warnsdorf's rule allows δ_r to be any coordinate in the interval $W_r = \bigl[\rho(r) \mathinner{.\,.} \rho'(r)\bigr)$, where $\rho'(r) = \rho\bigl(r \mathbin{\&} (r-1)\bigr)$ is the index of the next-to-rightmost 1 in the binary representation of r. (If r is a power of 2, $\rho'(r)$ is undefined; in such cases we use n instead of $\rho'(r)$ in the formula for W_r.) For example, if $r = (1010010)_2$ and $n \ge 7$, we have $\rho(r) = 1$ and $\rho'(r) = 4$, hence $W_r = [1 \mathinner{.\,.} 4) = \{1, 2, 3\}$.

Moreover, the residual graph of unvisited vertices, when it is time to choose δ_r, is always symmetrical with respect to every coordinate in W_r. Therefore we can choose $\delta_r = \rho(r)$ without loss of generality; all other Warnsdorf paths can be mapped into the

convex triangular grid graphs
SGB
simplex
dual
triangular faces
convex polygon
Gray binary code
delta sequence
ruler function ρ
Warnsdorf's rule
next-to-rightmost 1

standard path by permuting coordinates. For example, the Warnsdorf paths for the 3-cube have the delta sequences 0102010, 0102101, 0201020, 0201202, 1012101, 1012010, 1210121, 1210212, 2021202, 2021020, 2120212, 2120121.

[Algorithm W always succeeds in the graph $P_3 \mathbin{\square} P_3 \mathbin{\square} P_3$ when $s = (0,0,0)$; but not always in $P_4 \mathbin{\square} P_4 \mathbin{\square} P_4$. It sometimes fails in $P_3 \mathbin{\square} P_3 \mathbin{\square} P_3 \mathbin{\square} P_3$ when $s = (0,0,0,0)$.]

65. Yes: . (Is there a smaller one?)

70. It happens when $k = t - 1$ in the first call, and when $k = 2$ in the second call. (A little time can be saved by detecting these special cases. Similarly, '$\mathit{update}(u_1, \ldots, u_t)$' arises in step F5 when $k = j \pm 1$. Such updates weren't counted in the author's tests.)

71. Let the edges be $1 \mathbin{—} 2 \mathbin{—} \cdots \mathbin{—} n \mathbin{—} 1$ and $1 \mathbin{—} 5$, $3 \mathbin{—} (n{-}2)$, $(n{-}4) \mathbin{—} n$; consider the path $2 \mathbin{—} 1 \mathbin{—} 5 \mathbin{—} 4 \mathbin{—} 3 \mathbin{—} (n{-}2) \mathbin{—} (n{-}3) \mathbin{—} (n{-}4) \mathbin{—} n \mathbin{—} (n{-}1)$.

73. Assign a random 32-bit weight to each edge of the graph, and let each path have a "long hash code" H that's the sum of its edge weights (modulo 2^{32}). Let there be 2^b hash lists; a path with long hash H will go into list $H \bmod 2^b$. If c vertex names can be packed into an octabyte, the dictionary entry for $(v_1, \ldots, v_t)$ will occupy $B = 1 + \lceil t/c \rceil$ octabytes: one for the link and H, the others for the vertices (which are examined in detail during a search only when the long hash code is correct). Store the qth path in positions $(qB + s) \bmod M$ of a large array of octabytes, for $0 \le s < B$, where M is a large power of 2. (Overflow occurs if we try to write into position $(p_2 B + B) \bmod M$.)

75. (a) Let H be the graph whose vertices are the Hamiltonian paths of G that start with v_1, adjacent if they differ only by flipping a subpath. Since G is cubic, every vertex of H has degree 2. So H consists of cycles, and Algorithm F^- constructs the cycle that begins with the given path. For example, when $G = P_2 \mathbin{\square} P_2 \mathbin{\square} P_2$, we can represent the vertices (000, 001, …, 111) by (0, 1, …, 7), and H has two cycles: 01326754 — 01326457 — 01375462 — 02645731 — 02645137 — 02673154 — 04513762 — 04513267 — 04576231 — 01326754; 04623751 — 01573264 — 01546247 — 01546732 — 02376451 — 02315467 — 02315764 — 04675132 — 04623157 — 04623751.

(b) Let $\alpha = v_1 \mathbin{—} v_2 \mathbin{—} \cdots \mathbin{—} v_n$ be a Hamiltonian path with $v_n \mathbin{—} v_1$. One of its two neighbors in H is $v_1 \mathbin{—} v_n \mathbin{—} \cdots \mathbin{—} v_2$, which is the reflected cycle α^R. The other neighbor will have v_2 unchanged. So the cyclic paths come in pairs.

(c) Consider maximal segments of the cycle in H whose paths begin with $v_1 \mathbin{—} v_2$. The first and last of these paths are Hamiltonian cycles, which we can regard as *mates* of each other. (For example, the mate of 01326754 with respect to 0 — 1 is 01375462.)

(d) If α is a Hamiltonian cycle containing the edge e, its mate β is a Hamiltonian cycle containing an edge $e' \notin \alpha$. Hence γ, the mate of β with respect to e', is a third.

(e) Color the n edges of α alternately red and green; color the other $n/2$ edges blue. The blue edges of β and γ are the same; suppose there are $n/2 - x$ of them. Let β have r red and g green; hence γ has $n/2 - r$ red and $n/2 - g$ green. We have $r + g + n/2 - x = (n/2 - r) + (n/2 - g) + (n/2 - x) = n$; hence $x = 0$. Therefore no two consecutive edges of α can appear in β or γ; they must all be the same color.

[*Historical notes:* The theorem in (c) was discovered via algebraic reasoning by C. A. B. Smith, about 1940, but not published until later. See *Combinatorial Mathematics and its Applications* (Oxford conference, 1969), 259–283; W. T. Tutte, *Graph Theory As I Have Known It* (1998), 18, 48, 94. A. G. Thomason, in *Annals of Discrete Mathematics* **3** (1978), 259–268, introduced one-sided flips and used them to give an algorithmic proof of Smith's theorem.]

77. (a) Easily verified. (This is the *only* automorphism, when $n > 1$.)

Warnsdorf paths
author
random 32-bit weight
long hash code
3-cube
mates
Historical notes
Smith
Tutte
Thomason

Cameron
Krawczyk
Stappers
generating function
Zhong
Bondy
Murty

(b) Any Hamiltonian cycle containing $05 \mathrel{-\!\!-} 0 \mathrel{-\!\!-} 01$ mustn't contain $0 \mathrel{-\!\!-} 06$; hence $07 \mathrel{-\!\!-} 06 \mathrel{-\!\!-} 05$ but not $05 \mathrel{-\!\!-} 04$ or $01 \mathrel{-\!\!-} 07$; hence $03 \mathrel{-\!\!-} 04 \mathrel{-\!\!-} 15$, $01 \mathrel{-\!\!-} 02$, $08 \mathrel{-\!\!-} 07 \mathrel{-\!\!-} 06$, not $02 \mathrel{-\!\!-} 08$; hence $11 \mathrel{-\!\!-} 08$. Replacing all '$0j$' by '1' yields a Hamiltonian cycle containing $15 \mathrel{-\!\!-} 1 \mathrel{-\!\!-} 11$ in a graph isomorphic to C_{n-1}. By induction, it's α_n. Similarly, the only Hamiltonian cycles containing $06 \mathrel{-\!\!-} 0 \mathrel{-\!\!-} 01$ and $05 \mathrel{-\!\!-} 0 \mathrel{-\!\!-} 06$ are

$$\begin{aligned}\beta_n &= 0 \mathrel{-\!\!-} 01 \mathrel{-\!\!-} 07 \mathrel{-\!\!-} 08 \mathrel{-\!\!-} 02 \mathrel{-\!\!-} 03 \mathrel{-\!\!-} 16 \mathrel{-\!\!-} \cdots \mathrel{-\!\!-} 15 \mathrel{-\!\!-} 04 \mathrel{-\!\!-} 05 \mathrel{-\!\!-} 06 \mathrel{-\!\!-} 0;\\ \gamma_n &= 0 \mathrel{-\!\!-} 06 \mathrel{-\!\!-} 07 \mathrel{-\!\!-} 01 \mathrel{-\!\!-} 02 \mathrel{-\!\!-} 08 \mathrel{-\!\!-} 11 \mathrel{-\!\!-} \cdots \mathrel{-\!\!-} 16 \mathrel{-\!\!-} 03 \mathrel{-\!\!-} 04 \mathrel{-\!\!-} 05 \mathrel{-\!\!-} 0.\end{aligned}$$

(c) (11, 65, 265, 1005, 3749, 13927, 51683, 191735, 711243). [But an appropriate sequence of only $4n$ *two-sided* flips will take us from α_n to β_n, or β_n to γ_n, or γ_n to α_n.]

(d) When $n > 2$, the first five flips yield $01 \mathrel{-\!\!-} 0 \mathrel{-\!\!-} 05 \mathrel{-\!\!-} 06 \mathrel{-\!\!-} 07 \mathrel{-\!\!-} 08 \mathrel{-\!\!-} 02 \mathrel{-\!\!-} 03 \mathrel{-\!\!-} 04 \mathrel{-\!\!-} 15 \mathrel{-\!\!-} 16 \mathrel{-\!\!-} 17 \mathrel{-\!\!-} 11 \mathrel{-\!\!-} 12 \mathrel{-\!\!-} 18$ followed by a sequence $21 \mathrel{-\!\!-} 22 \mathrel{-\!\!-} \cdots$ that's the same as the suffix $01 \mathrel{-\!\!-} 02 \mathrel{-\!\!-} \cdots$ of the second path obtained with respect to $0 \mathrel{-\!\!-} 01$, *except* that all entries are increased by 20, and '$14 \mathrel{-\!\!-} 13$' appears between 25 and 26. The next $c_{n-2} - 1$ flips mimic (c); then six more flips give the reverse of β_n.

(e) When $n > 1$, the number is $c_{n-1} + 5$ in both cases(!), proved as in (d).

[The graphs C_n were introduced by K. Cameron, *Discrete Math.* **235** (2001), 69–77, who simplified a similar construction by A. Krawczyk and proved that $c_n \geq 2^n$. In 2020, Filip Stappers discovered that the generating function $c(z) = \sum_n c_n z^n$ is $p(z)/q(z)$, where $q(z) = (1-z)(1-3z-2z^2-2z^3-z^4-z^5)$ and $p(z) = z(1+z)(11+10z+6z^2+4z^3+z^4)$. He also proved that the number of one-sided flips to go from β_n to its mate γ_n, with respect to either $0 \mathrel{-\!\!-} 06$ or $06 \mathrel{-\!\!-} 0$, is $\hat{c}_n$, where $\sum_n \hat{c}_n z^n = \hat{p}(z)/q(z)$ and $\hat{p}(z) = 2z(3+2z+z^2-2z^3)$. Consequently the actual limiting ratio c_{n+1}/c_n is $\rho \approx 3.709398$, the real root of $z^5 = 3z^4+2z^3+2z^2+z+1$. Asymptotically, $c_n \sim c\rho^n - 8$ and $\hat{c}_n \sim \hat{c}\rho^n$, where $c \approx 5.349$ and $\hat{c}_n/c_n \sim \hat{p}(\rho)/p(\rho) \approx 0.3959$. In *Bull. Aust. Math. Soc.* **98** (2018), 18–26, L. Zhong introduced a family of graphs on $16n$ vertices for which the number of flips to get from a certain Hamiltonian path to its mate with respect to $0 \mathrel{-\!\!-} 1$ is *exactly* $6 \cdot 2^n - 10$. However, that number with respect to $1 \mathrel{-\!\!-} 0$ is only 4(!).]

78. (a, b) In contrast to exercise 60, success occurs with probability $\approx 100\%$ when $q \leq 10$. Furthermore, a Hamiltonian cycle is usually found soon after finding the first Hamiltonian path. The average number of updates before that first cycle, observed in 100 runs for each q, was $\approx$ (81, 141, 146, 240, 295).

But $q = 100$ was a different story. Here a 400-cycle was successfully found in only six of ten cases — sometimes after as few as 18 thousand updates, sometimes after as many as 8.3 million. In one of the other cases, millions of Hamiltonian paths (not cycles) were found; but memory overflow, with more than 2 million paths in the dictionary, aborted the run. Memory overflow also arose in the three other cases, once before achieving any paths longer than 365.

We conclude that Algorithm F can have wildly eccentric behavior, and it should probably be restarted if it spins its wheels too long.

79. (a) Let there be 12 vertices $\{0, \ldots, 11\}$, with $k \mathrel{-\!\!-} (k{+}1)$ for $0 \leq k < 12$ and $3k \mathrel{-\!\!-} (3k{+}5)$ for $0 \leq k < 4$ (modulo 12). This graph has two equivalence classes, each containing one Hamiltonian cycle and 12 Hamiltonian paths that aren't cycles.

(b) Let there be $13n$ vertices ij for $0 \leq i < n$ and $-1 \leq j < 12$, with $ik \mathrel{-\!\!-} ik'$ whenever $k \mathrel{-\!\!-} k'$ in (a); also $i\bar{1} \mathrel{-\!\!-} i0$ and $i1 \mathrel{-\!\!-} ((i{+}1) \bmod n)\bar{1}$. This graph has 2^n equivalence classes, each containing one cycle and $17n$ noncycles.

80. If and only if $s \leq t$ (except when $s = t = q_1 = 1$). [See J. A. Bondy and U. S. R. Murty, *Graph Theory* (2008), Theorem 18.1.]

81. (a) Choose nonadjacent $\{u,v\}$ with $\deg(u) \le \deg(v)$ so that $\deg(u)+\deg(v)$ is maximum, and assume that $\deg(u)+\deg(v) < n$. Let $k = \deg(u)$. Then $k > 0$, because there are no isolated vertices when $d_{n-1} = n-1$. Exactly $n-1-\deg(v) \ge k$ vertices $\ne v$ are nonadjacent to v; these must all have degree $\le k$, by maximality. Similarly, exactly $n-k \ge \deg(v)$ vertices are nonadjacent to u, and they all have degree $\le \deg(v)$.

But $d_s \le t$ if and only if at least s vertices have degree $\le t$. Hence we have proved that $1 \le k < n/2$, $d_k \le k$, and $d_{n-k} \le \deg(v) < n-k$, contradicting $(*)$.

(b) Each G_k satisfies $(*)$, so G_{k+1} exists. Let $(w_0 w_1 \ldots w_{n-1})$ be a cycle in G_{k+1} that's not also a cycle in G_k. We can assume that $w_0 = u_k$ and $w_{n-1} = v_k$. There are $\deg(u_k)$ values of j with $w_0 \mathrel{-\!\!-} w_{j+1}$ in G_k. And $w_{n-1} \mathrel{-\!\!-} w_j$ for at least one such j, because $\deg(w_{n-1}) \ge n - \deg(w_0)$. Thus $(w_0 \ldots w_j w_{n-1} \ldots w_{j+1})$ is a cycle in G_k.

[Condition $(*)$ was discovered by V. Chvátal, *J. Comb. Theory* **12** (1972), 163–168. This proof is due to J. A. Bondy and V. Chvátal, *Discrete Math.* **15** (1976), 111–135.]

82. Let G' be the graph $(kK_1 \oplus K_{n-2k}) \mathrel{-\!\!-} K_k$, whose degree sequence has $d'_j = k$ for $0 < j \le k$, $d'_j = n-1-k$ for $k < j \le n-k$, $d'_j = n-1$ for $n-k < j \le n$. (See exercise 80.)

83. (a) There are $(2r+1)r/2$ edges. (b) Use exercises 2 and 81.

(c) If $u_0 \mathrel{-\!\!-} u_j$ and $u_{j-1} \mathrel{-\!\!-} u_{2r}$, a flip will create a cycle. So r vertices u_{j-1} cannot be adjacent to u_{2r}; the remaining r candidates must be u_{2r}'s neighbors.

(d) If the neighbors of u_0 are $u_1, \ldots, u_r$ and the neighbors of u_{2r} are $u_r, \ldots, u_{2r-1}$, we have a $(2r+1)$-cycle. Otherwise let j be minimum such that $u_0 \mathrel{-\!\!\!/\!\!-} u_j$ and $u_0 \mathrel{-\!\!-} u_{j+1}$. Then $u_{j-1} \mathrel{-\!\!-} u_{2r}$, and we have a $2r$-cycle that excludes $v_0 = u_j$.

(e) Assuming the hint, we can make a cycle $v_1 \mathrel{-\!\!-} \cdots \mathrel{-\!\!-} v_{2k-1} \mathrel{-\!\!-} v_0 \mathrel{-\!\!-} v_{2k+1} \mathrel{-\!\!-} \cdots \mathrel{-\!\!-} v_1$ that excludes v_{2k}, for any k; hence $v_{2k} \mathrel{-\!\!-} v_j$ for all odd j. But then v_1 has degree $r+1$. [This result was announced in *Lecture Notes in Math.* **186** (1971), 201.]

Notice that Hamiltonicity is *not* implied by exercise 81, even though that exercise is "best possible" according to exercise 82. No efficient way is known to test whether all graphs with a given degree sequence are forcibly Hamiltonian.

84. Let $t = \lceil n/2 \rceil$, and consider 2^{t-2} cases $a_1 \ldots a_t$ where $a_1 = a_t = 0$ and $a_k = [d_k \le k]$ for $1 \le k < t$. Then it's easy to see that the minimum $d_1 + \cdots + d_n$ in case $a_1 \ldots a_t$ occurs when $k < t$ and $a_k = 0$ implies $d_k = k+1$, $d_{n-k} = d_{n-k-1}$; $a_k = 1$ implies $d_k = d_{k-1}$, $d_{n-k} = n-k$; also $d_n = d_{n-1}$. (For example, if $n = 11$ and $a_1 \ldots a_6 = 010110$, we have $d_1 \ldots d_{11} = 22444677999$.) Let $s(a_1 \ldots a_t)$ denote this minimum sum.

Suppose j is minimum with $a_j = 1$, and k is minimum with $k > j$ and $a_k = 0$. One can show without difficulty that $s(0^{k-1} a_k \ldots a_t) < s(0^{j-1} 1^{k-j} a_k \ldots a_t)$, except that the inequality is reversed when n is odd and $j = t-1$. Consequently the overall minimum sum occurs uniquely for $d_1 \ldots d_n = 23 \ldots (t-1) t^{t+2}$ when n is even, $23 \ldots (t-1)(t-1) t^t$ when $n > 3$ is odd. Increase d_n by 1 if the sum is odd.

The resulting sequence of degrees is graphical, by exercise 7–105. Hence the answer turns out to be $\lfloor (3n^2 + 6n)/16 \rfloor$ when n is even; $\lfloor (3n^2 + 8n - 3)/16 \rfloor$ when n is odd.

85. The quadratic function $f(n,k)$ satisfies $f(n,k) \ge f(n,k+1)$ if and only if $k < \frac{n-1}{3}$. Thus $g(n,k) = \max_{k \le t < n/2} f(n,t)$. Every graph $(tK_1 \oplus K_{n-2t}) \mathrel{-\!\!-} K_t$ for $k \le t < n/2$ is non-Hamiltonian, with degree sequence $t^t (n-1-t)^{n-2t} (n-1)^t$ and $f(n,t)$ edges. Furthermore, every graph with $d_t \le t$ has at most t^2 edges that involve its first t vertices and at most $\binom{n-t}{2}$ edges that don't. Hence a graph with $d_1 \ge k$ and more than $g(n,k)$ edges must have $d_t > t$ for $k \le t < n/2$. And exercise 81 calls it Hamiltonian. [*Magyar Tudományos Akadémia Matematikai Kutató Int. Közl.* **7** (1962), 227–228.]

86. Every graph $\bigl((t+1)K_1 \oplus K_{n-1-2t}\bigr) \mathrel{-\!\!-} K_t$, for $k \le t < (n-1)/2$, is untraceable. So we can achieve $\hat{g}(n,k) = \max\bigl(f(n,k), f(n, \lfloor n/2 \rfloor - 1)\bigr)$ edges, when $0 \le k < \lfloor n/2 \rfloor$.

isolated vertices
Chvátal
Bondy
flip
degree sequence
forcibly Hamiltonian
graphical
degree sequence

components
articulation point
components

On the other hand, by exercises 2 and 81, a graph is traceable whenever its degree sequence $d_1 \le \cdots \le d_n$ satisfies the following condition:

$$1 \le t < (n+1)/2 \ \text{ and } \ d_t < t \ \text{ implies } d_{n+1-t} \ge n-t. \qquad (+)$$

In particular, a graph with minimum degree $d_1 \ge \lfloor n/2 \rfloor$ is always traceable. If $k < \lfloor n/2 \rfloor$ and $(+)$ fails for some t, we have $d_s \le t-1$ for $1 \le s \le t$; $d_s \le n-t-1$ for $t < s \le n+1-t$; and $d_s \le n-1$ for $n+1-t < s \le n$. Hence $(d_1 + \cdots + d_n)/2 \le \hat{f}(n, t-1) \le \hat{g}(n,k)$. (The last inequality holds because $k \le t-1 \le \lfloor n/2 \rfloor - 1$.)

88. (a) Let $v_0 \mathrel{\text{---}} \cdots \mathrel{\text{---}} v_l$ be a longest path, and assume that $l < 2k$. We will prove first that there's actually an l-cycle, using the fact that all neighbors of v_0 and v_l must lie on that path. Indeed, let $\{v_i \mid i \in I\}$ be the neighbors of v_0, and let $\{v_{j-1} \mid j \in J\}$ be the neighbors of v_l. Then I and J are subsets of $\{1, \ldots, l\}$. They can't be disjoint, because $|I| \ge k$ and $|J| \ge k$. Therefore there's some $j \in I \cap J$; and we have the cycle $v_0 \mathrel{\text{---}} \cdots \mathrel{\text{---}} v_{j-1} \mathrel{\text{---}} v_l \mathrel{\text{---}} \cdots v_j \mathrel{\text{---}} v_0$.

But there can't *be* an l-cycle! Since $l \le n-2$ and G is connected, there must be vertices w and w' not on the cycle, with $v_j \mathrel{\text{---}} w \mathrel{\text{---}} w'$ for some j. So there's a longer path.

(b) The result clearly holds for $n \le l+1$, because the number of edges is $\le \binom{n}{2} \le nl/2$. Also for larger n, if G isn't connected; for if there are r components, with n_j vertices and m_j edges in component j, each n_j is less than n. By induction, the number $m_1 + \cdots + m_r$ of edges is at most $(n_1 + \cdots + n_r)l/2 = nl/2$.

Assume therefore that $n > l+1$ and G is connected. Let $k = \lfloor l/2 \rfloor + 1$. Then $2k > l$, so there's no path of length $2k$. Hence by (a), there's a vertex v of degree $< k$, unless $n = 2k = l+2$. And v exists even in that case; otherwise exercise 81 tells us there would be a cycle of length $2k$, hence a path of length $2k-1 > l$.

Now $G \setminus v$ has at most $(n-1)l/2$ edges; so G has at most $(n-1)l/2 + k - 1 \le nl/2$.

(c) $\lfloor n/(l+1) \rfloor K_{l+1} \oplus K_{n \bmod (l+1)}$, a graph with $\lceil n/(l+1) \rceil$ components. (The same number of edges is achieved by the much more interesting graph $K_{l/2} \mathrel{\text{---}} \overline{K}_{n-l/2}$, if l is even and $n > l$ and $n \bmod (l+1) \in \{l/2, l/2+1\}$!)

89. (a) Let l be the length of G, and consider a longest path $v_0 \mathrel{\text{---}} v_1 \mathrel{\text{---}} \cdots \mathrel{\text{---}} v_l$ where $v_l \mathrel{\text{---}} v_p$ and p is as small as possible. The resulting cycle has length $c = l+1-p$; so we assume that $c < 2k$. A vertex v_q will be called "bounded" if its neighbors all belong to the cycle. We shall prove that v_q is bounded whenever $p < q \le l$.

The idea will be to construct a longest path $v_0 \mathrel{\text{---}} \cdots \mathrel{\text{---}} v_p \mathrel{\text{---}} v'_{p+1} \mathrel{\text{---}} \cdots \mathrel{\text{---}} v'_l$, where $\{v'_{p+1}, \ldots, v'_l\} = \{v_{p+1}, \ldots, v_l\}$ and $v'_l = v_q$. Then v_q must be bounded, because l is maximum and p is minimum. Vertex v_l is clearly bounded; so is vertex v_{p+1}.

Suppose v_{q+1} is bounded, and let the neighbors of v_{p+1} and v_{q+1} be $\{v_i \mid i \in I\}$ and $\{v_j \mid j \in J\}$. Then $I \cup J \subseteq \{v_{p+1}, \ldots, v_{l+1}\}$, where we set $v_{l+1} = v_p$. Also $|I|, |J| \ge k$.

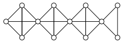
If $i \in I$ and $i \le q$ and $i-1 \in J$, let $v_p v'_{p+1} \ldots v'_l = v_{l+1} \ldots v_{q+1} v_{i-1} \ldots v_{p+1} v_i \ldots v_q$. If $i \in I$ and $q < i \le l$ and $i+1 \in J$, let $v_p v'_{p+1} \ldots v'_l = v_{l+1} \ldots v_{i+1} v_{q+1} \ldots v_i \mathrel{\text{---}} v_{p+1} \ldots v_q$. One of these constructions must work; otherwise we'd have ruled out at least $k-1$ of the c potential elements of J, and we also have $q+1 \notin J$.

But $\{v_{p+1}, \ldots, v_l\}$ can't all be bounded! If $p = 0$, the graph G would be disconnected; otherwise vertex v_p would be an articulation point.

(b) The result clearly holds for $n \le c$, because the number of edges is $\le \binom{n}{2} \le (n-1)c/2$. Also for larger n, if G isn't connected; for if there are r components, with n_j vertices and m_j edges in component j, each n_j is less than n. By induction, the number $m_1 + \cdots + m_r$ of edges is at most $((n_1 - 1) + \cdots + (n_r - 1))c/2 < (n-1)c/2$.

Assume therefore that $n > c$ and G is connected. If G isn't biconnected, there's an articulation point v that divides G into a bicomponent G' containing v and a connected graph $(G \setminus G') \cup v$. If G' has n' vertices, G has $\le (n'-1)c/2 + (n-n')c/2$ edges.

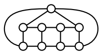
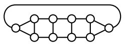
Finally, assume that G is biconnected and $n > c$. The proof follows as in exercise 88(b), because there exists a vertex whose degree is less than $k = \lfloor c/2 \rfloor + 1$.

(c) $K_1 \mathrel{-\!\!-\!\!-} (\lfloor (n-1)/(c-1) \rfloor K_{c-1} \oplus K_{(n-1) \bmod (c-1)})$. (The same number of edges is achieved by a traceable graph: Put $\lfloor (n-1)/(c-1) \rfloor$ copies of K_c and a $K_{1+(n-1) \bmod (c-1)}$ in a row, then paste them together;  if $(n,c) = (12,4)$.)

Historical notes: These results and those of the previous exercise are due to P. Erdős and T. Gallai [*Acta Mathematica Academiæ Scientiarum Hungaricæ* **10** (1959), 337–356]. R. J. Faudree and R. H. Schelp [*J. Combinatorial Theory* **B19** (1975) 150–160] proved that the lower bound of exercise 88(c) is sharp: The upper bound in 88(a) can be replaced by the size of those graphs. Similarly, D. R. Woodall [*Acta Math. Acad. Sci. Hung.* **28** (1976), 77–80] proved that the lower bound in (c) is sharp.

90. True, except when G has no edges (and length 0). See exercise 2.

93. (a) True, unless there are fewer than 4 vertices.

(b) Graphs like  and  for $n = 9$ and $n = 10$ work in general.

[*Mathematical Gazette* **49** (1965), 40–41. These cubic graphs for even n are also perfectly Hamiltonian. A more symmetrical graph, whose edges are $k \mathrel{-\!\!-\!\!-} (k+1)$ and $k \mathrel{-\!\!-\!\!-} (k+n/2)$ (modulo n), can also be used when n is a multiple of 4.]

95. (a) Powers of the "obvious" permutation $\sigma = (a_0a_1a_2a_3a_4a_5a_6)(b_0b_1b_2b_3b_4b_5b_6)$ $(c_0c_1c_2c_3c_4c_5c_6)(d_0d_1d_2d_3d_4d_5d_6)$ will take $d_6 \mapsto d_j$ for any j. There's also a "surprise," $\rho = (a_0b_0)(a_1b_2)(a_2d_2)(a_3c_2)(a_4c_5)(a_5d_5)(a_6b_5)(b_1b_6)(b_3d_6)(b_4d_1)(c_1d_4)(c_6d_3)$; one can verify that $u\rho \mathrel{-\!\!-\!\!-} v\rho$ whenever $u \mathrel{-\!\!-\!\!-} v$. (Notice that c_0, c_3, c_4, and d_0 are fixed by ρ. Coxeter called this "an apparent miracle.") When ρ is premultiplied and postmultiplied by appropriate powers of σ, we can take d_0 into any desired vertex.

The mapping $a_j \mapsto b_{5j}$, $b_j \mapsto c_{5j}$, $c_j \mapsto a_{5j}$, $d_j \mapsto d_{5j}$, namely the permutation $\tau = (a_0b_0c_0)$ $(a_1b_5c_4a_6b_2c_3)$ $(b_1c_5a_4b_6c_2a_3)$ $(c_1a_5b_4c_6a_2b_3)$ $(d_1d_5d_4d_6d_2d_3)$, is another automorphism that fixes d_0. When d_0 is fixed, we must take its neighbor c_0 into a neighbor; hence we can let $S_{26} = \{(), \tau, \tau^2\}$. And when c_0 is also fixed, we can let $S_{25} = \{(), \rho\}$, because b_0 must map to itself or a_0. Clearly $S_{24} = \{()\}$.

Finally, can we move anything else when d_0, c_0, b_0, a_0 are all fixed? Aha — there's just one possibility, namely τ^3, which swaps $a_j \leftrightarrow a_{-j}$, $b_j \leftrightarrow b_{-j}$, $c_j \leftrightarrow c_{-j}$, $d_j \leftrightarrow d_{-j}$, for $0 < j < 7$. Thus $S_{23} = \{(), \tau^3\}$ and $S_{22} = \cdots = S_1 = \{()\}$.

(b) Part (a) explains how to map $v \mapsto d_0 \mapsto v'$.

(c) In fact, part (a) shows that $u \mathrel{-\!\!-\!\!-} v \mathrel{-\!\!-\!\!-} w$ can be mapped to any $u' \mathrel{-\!\!-\!\!-} v' \mathrel{-\!\!-\!\!-} w'$.

(d) Algorithm H quickly shows that that there are no Hamiltonian cycles. But there are 12, such as $a_1 \mathrel{-\!\!-\!\!-} a_0 \mathrel{-\!\!-\!\!-} a_6 \mathrel{-\!\!-\!\!-} a_5 \mathrel{-\!\!-\!\!-} a_4 \mathrel{-\!\!-\!\!-} a_3 \mathrel{-\!\!-\!\!-} a_2 \mathrel{-\!\!-\!\!-} d_2 \mathrel{-\!\!-\!\!-} c_2 \mathrel{-\!\!-\!\!-} c_6 \mathrel{-\!\!-\!\!-} d_6 \mathrel{-\!\!-\!\!-} b_6 \mathrel{-\!\!-\!\!-} b_1 \mathrel{-\!\!-\!\!-} b_3 \mathrel{-\!\!-\!\!-} d_3 \mathrel{-\!\!-\!\!-} c_3 \mathrel{-\!\!-\!\!-} c_0 \mathrel{-\!\!-\!\!-} c_4 \mathrel{-\!\!-\!\!-} d_4 \mathrel{-\!\!-\!\!-} b_4 \mathrel{-\!\!-\!\!-} b_2 \mathrel{-\!\!-\!\!-} b_0 \mathrel{-\!\!-\!\!-} b_5 \mathrel{-\!\!-\!\!-} d_5 \mathrel{-\!\!-\!\!-} c_5 \mathrel{-\!\!-\!\!-} c_1 \mathrel{-\!\!-\!\!-} d_1 \mathrel{-\!\!-\!\!-} a_1$, that omit (say) d_0.

Historical notes: The Coxeter graph was first discussed in print by W. T. Tutte [*Canadian Math. Bulletin* **3** (1960), 1–5], who proved it non-Hamiltonian. Eventually H. S. M. Coxeter wrote about "his graph" [*J. London Math. Society* (3) **46** (1983), 117–136], identifying its vertices with the $\binom{8}{2} = 28$ unordered pairs $\{x, y\}$ of the set $D = \{0, 1, 2, 3, 4, 5, 6, \infty\}$. His new names for vertices a_0 through d_7 were respectively 25, 36, 04, 15, 26, 03, 14; 34, 45, 56, 06, 01, 12, 23; 16, 02, 13, 24, 35, 46, 05; 0∞, 1∞, 2∞, 3∞, 4∞, 5∞, 6∞ (abbreviating $\{x, y\}$ by xy). If $0 \le x < y < 7$, the neighbors

articulation point
bicomponent
traceable
Historical notes
Erdős
Gallai
Faudree
Schelp
Woodall
cubic graphs
perfectly Hamiltonian
Coxeter
miracle
Historical notes
Tutte
Coxeter

of xy are $\{2x-y, 3x-2y\}$, $\{2y-x, 3y-2x\}$, and $\{4x+4y, \infty\}$, using arithmetic mod 7. He showed that the $7^3-7=336$ automorphisms correspond to the mappings $\{x,y\} \mapsto \{f(x), f(y)\}$, where f is a fractional linear transformation on D; that is, $f(x)=(ax+b)/(cx+d)$, where $0 \le a,b,c,d < 7$ and $(ad-bc) \bmod 7 \ne 0$ and either $c=1$ or $(c,d)=(0,1)$. (In this computation, $x/\infty=0$, $x/0=\infty$, and $f(\infty)=a/c$. The automorphisms σ, ρ, τ above correspond respectively to $f(x)=x+1$, $1/x$, $5x$.)

100. (Using ideas of N. Beluhov.) When C is a cycle cover, let $s_j = 4[t_j \mathrel{-\!\!-} t_{j+1} \in C] + 2[v_j \mathrel{-\!\!-} w_{j+1} \in C] + [w_j \mathrel{-\!\!-} v_{j+1} \in C]$ encode its edges between indices j and $j+1$ modulo q. A simple case analysis shows that $s_j \ne 0$; $s_j \in \{1,2,4\} \implies s_{j+1}=7$; $s_j=3 \implies s_{j+1} \in \{5,6\}$; $s_j=5 \implies s_{j+1} \in \{3,5\}$; $s_j=6 \implies s_{j+1} \in \{3,6\}$; $s_j=7 \implies s_{j+1} \in \{1,2,4\}$; and that the sequence $s_1 s_2 \ldots s_q$ completely determines C.

Thus there are two kinds of covers: Type A, where s_j is alternately 7 and an element of $\{1,2,4\}$; or type B, where each s_j is an element of $\{3,5,6\}$. Type A covers arise only when q is even, and they have $k+1$ cycles when there are k occurrences of $s_j = s_{j+2} \ne 7$. Type B covers always have exactly 2 cycles.

Let $g(w,z) = \sum w^{[a_0=a_1]+\cdots+[a_{n-1}=a_n]} z^n [a_0=a_n]$, summed over all ternary sequences $a_0 a_1 \ldots a_n$, and let $h(w,z)$ be similar but requiring $a_0 \ne a_n$. Then $g(w,z) = 3 + wzg(w,z) + zh(w,z)$ and $h(w,z) = 2zg(w,z) + (1+w)zh(w,z)$. So we find $g(w,z) = 3(1-(1+w)z)/((1-(w-1)z)(1-(w+2)z)) = 2/(1-(w-1)z) + 1/(1-(2+w)z)$. Consequently the number of type A covers with k cycles is $2\,[w^{k-1}z^{q/2}]\,g(w,z) = 4\binom{q/2}{k-1}(2^{q/2-k} - (-1)^{q/2-k})$ when q is even. (In particular, the number of Hamiltonian cycles is $4(2^{q/2-1} + (-1)^{q/2})$.)

Turning to type B, let there be f_{xyn} sequences $a_0 \ldots a_n$ with $a_0 = x$, $a_n = y$, and each $a_j \in \{3,5,6\}$, having no consecutive 33 or 56 or 65. We find by induction that $f_{xyn} = (2^n - (-1)^n)/3 + \delta_{xyn}$, where $\delta_{xyn} = 1$ when n is even and $x = y$, $\delta_{xyn} = -1$ when n is odd and $xy \in \{33, 56, 65\}$, otherwise $\delta_{xyn} = 0$. Hence there are $f_{33q} + f_{55q} + f_{66q} = 2^q + 2[q \text{ even}]$ covers of type B.

103. Let $H = v_0 \mathrel{-\!\!-} v_1 \mathrel{-\!\!-} \cdots \mathrel{-\!\!-} v_n = v_0$. Every edge of $G \setminus H$ has the form $e_{ij} = v_i \mathrel{-\!\!-} v_j$ for some $0 \le i < j < n$. When G is drawn in the plane with no crossing edges, two edges e_{ij} and $e_{i'j'}$ with $i < i' < j < j'$ cannot both lie inside H, nor can they both lie outside H. Therefore the graph E whose vertices are the e_{ij}, with e_{ij} adjacent to $e_{i'j'}$ when $i < i' < j < j'$, must be bipartite. Conversely, if E is bipartite, G is clearly planar. (And bipartiteness is readily tested by Algorithm 7B.)

Historical notes: This criterion for planarity was discovered by G. Demoucron, Y. Malgrange, and R. Pertuiset [*Revue Française de Recherche Opérationnelle* **8** (1964), 33–47] and independently by W. Bader [*Archiv für Elektrotechnik* **49** (1964), 2–12], at the time when planarity of printed circuits began to be important. A graph is planar if and only if its blocks (biconnected components) are planar; and in practice, a block that isn't a single edge is almost always Hamiltonian. Notice that the nonplanar graphs K_5 and $K_{3,3}$ have Hamiltonian cycles, and the corresponding graphs E aren't 2-colorable.

105. (a) $[n \ge 3]\,n!/(2n)$, one for every pair $\{\pi, \pi^-\}$ where π is a cyclic permutation.
(b) $[m = n \ge 2]\,n!^2/(2n)$.

106. [This problem is scheduled to appear in the *American Mathematical Monthly*, so the answer is "embargoed" until the deadline for reader submissions has passed.]

108. (a) Those are the only remaining ways to include vertex 28 in the cycle.
(b) $\genfrac{}{}{0pt}{}{08}{16}$ would form a short cycle; so 08 must be covered by $\genfrac{}{}{0pt}{}{08}{27}$.
(c) 14 must be covered by $\genfrac{}{}{0pt}{}{06}{14}$ and $\genfrac{}{}{0pt}{}{14}{26}$; but then $26 \mathrel{-\!\!-} 14 \mathrel{-\!\!-} 06 \mathrel{-\!\!-} \cdots \mathrel{-\!\!-} 26$.

fractional linear transformation
Beluhov
generating functions
ternary sequences
bipartite
Historical notes
Demoucron
Malgrange
Pertuiset
Bader
printed circuits
blocks
biconnected components
K_5
$K_{3,3}$
2-colorable
cyclic permutation

(d) We must choose $\genfrac{}{}{0pt}{}{14}{22}$ to avoid the contradiction, after which $\genfrac{}{}{0pt}{}{03}{15}$ leaves 23 — 04 — 25 and 05 — 24 — 16 as the only ways to cover 04 and 24. Then $\genfrac{}{}{0pt}{}{07}{15}$ is forced, and almost everything is nailed down. Hence there are two ways to complete a cycle after the choices of (a), (b), and (d): either $\{\genfrac{}{}{0pt}{}{05}{13}, \genfrac{}{}{0pt}{}{06}{25}, \genfrac{}{}{0pt}{}{14}{26}\}$ or $\{\genfrac{}{}{0pt}{}{05}{26}, \genfrac{}{}{0pt}{}{06}{14}, \genfrac{}{}{0pt}{}{13}{25}\}$.

NAME(v)
complete tree
rotations
binary trees
associahedron
"net" graph
Bauer
Broersma
Veldman
Chvátal

109. (In this graph we have `NAME(0)` = 00, `NAME(1)` = 01, ..., `NAME(29)` = 29.) `MATE(0)` is ≥ 0; `MATE(1)` $= 21$; `MATE(2)` $= 22$; `MATE(3)` $= 23$; `MATE(4)` $= -1$.

111. `TRIG` needs at most n locations, because no vertex can be a trigger more than once (when its degree drops to 2). `ACTIVE` needs at most $\binom{n+1}{2}$ locations, because at most $n-l$ vertices are outer in level l. `SAVE` needs at most n^2 slots, where each "slot" holds a mate and a degree, because there can be at most n levels.

112. (a) activate(u); activate(w); remarc(u,v); remarc(w,v); and makemates(u,w).

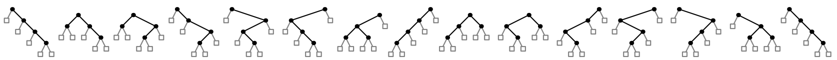
(b) activate(u); remarc(u,v); purge(w,v); makemates(`MATE(`w`)`$,u$); deactivate(w). Here 'purge(w,v)' means 'remarc(`NBR[`w`][`k`]`$,w$) for k decreasing from `DEG(`w`)`-1 down to 0, except when `NBR[`w`][`k`]` $= v$'.

(c) activate(w); remarc(w,v); purge(u,v); makemates(`MATE(`u`)`$,w$); deactivate(u).

(d) Do nothing if $e = n$. Otherwise purge(u,v); purge(w,v); makemates(`MATE(`u`)`, `MATE(`w`)`); deactivate(u); deactivate(w).

113. Set $x[k] \leftarrow -1$ for $0 \leq k < n$. Then do this for $0 \leq k < n$: If $x[\texttt{EU[}k\texttt{]}] < 0$, set $x[\texttt{EU[}k\texttt{]}] \leftarrow \texttt{EV[}k\texttt{]}$; else set $y[\texttt{EU[}k\texttt{]}] \leftarrow \texttt{EV[}k\texttt{]}$. If $x[\texttt{EV[}k\texttt{]}] < 0$, set $x[\texttt{EV[}k\texttt{]}] \leftarrow \texttt{EU[}k\texttt{]}$; else set $y[\texttt{EV[}k\texttt{]}] \leftarrow \texttt{EU[}k\texttt{]}$. Finally set $v_1 \leftarrow 0$, $v_2 \leftarrow x[0]$, and for $3 \leq k \leq n$ set $v_k \leftarrow (v_{k-2} = x[v_{k-1}]?\ y[v_{k-1}]\colon x[v_{k-1}])$.

115. Assume that $n > 4$. The root has degree $n-2$. The kth subtree, for $1 \leq k < n-2$, is $T(n-1-k, n-3, n-2, \ldots, 2)$; and the last subtree is $T(n-3, n-2, \ldots, 2)$. Here $T(d_0, \ldots, d_{r-1})$ denotes the complete tree with d_l-way branching at level l for $0 \leq l < r$. (The kth subtree has $(n-1-k)(n-3)!$ of the $(n-1)!/2$ solutions.)

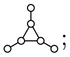
116. 
is one of the six ways to do 14 suitable rotations of those binary trees. (These are also the Hamiltonian cycles of an *associahedron*; see exercise 7.2.1.6–29.)

117. (a) True. ($t(G) = 0$ if and only if $U = \emptyset$ disconnects G.)

(b) If $|U|/k(G\backslash U) \neq |U|/k(G\backslash e\backslash U)$, edge e joins two components of $G\backslash e$; hence $k(G\backslash U) = k(G\backslash e\backslash U) - 1$ (and the term for this U leaves the 'min' if $k(G\backslash U) = 1$).

(c) This follows the monoticity proved in (b), since $t(C_n) \geq 1$.

(d) After cutting out m' vertices of the smaller part and n' vertices of the larger part, the residual graph that's left is connected unless $m' = m$ or $n' = n$. The smallest ratio $(m'+n')/(m-m'+n-n')$ is m/n in such cases. (See exercise 105.)

(e) 4/3, by cutting 4 independent vertices. (Let N be the "net" graph, ; the smallest non-Hamiltonian tough graph is K_1 — N. A 42-vertex non-Hamiltonian graph with $t(G) = 2$ was (surprisingly) constructed by D. Bauer, H. J. Broersma, and H. J. Veldman, in *Discrete Applied Math.* **99** (2000), 317–321; it's tough for Algorithm H!)

(Let $N_{t,m}$ be the graph $(t+1)K_m$ — K_t; graph D in Table 1 is the special case $N_{5,3}$. We have $t(N_{t,m}) \leq t/(t+1)$; hence $N_{t,m}$ is non-tough. Chvátal's original paper about toughness appeared in *Discrete Mathematics* **5** (1973), 215–228.)

118. (Solution by V. Chvátal.) Let the vertices of $G = K_m \,\square\, K_n$ be $V \times W$, where $|V| = m$ and $|W| = n$. Given $p > 1$, the smallest set U that makes $k(G \setminus U) = p$ has the form $(V \times W) \setminus (V_1{\times}W_1 \cup \cdots \cup V_p{\times}W_p)$, where $V_1 \cup \cdots \cup V_p$ and $W_1 \cup \cdots \cup W_p$ are set partitions of V and W. Hence $|U| = mn - m_1n_1 - \cdots - m_pn_p$, where the sizes

$|V_j| = m_j$ and $|W| = n_j$ are positive integers that sum to m and n. It is minimized when $m_1 = m + 1 - p$, $n_1 = n + 1 - p$, and all other m_j and n_j are 1; in other words, the smallest such $|U|$ is $mn - (p{-}1) - (m{+}1{-}p)(n{+}1{-}p) = (p-1)(m+n-p)$. Hence $t(G) = \min_{p=2}^{\min(m,n)}(p-1)(m+n-p)/p = (m+n-2)/2$.

119. They take $v \mapsto \bigl(\frac{av+b}{cv+d}\bigr) \bmod 47$, where $(a,b,c,d) = (20,15,17,27)$, $(20,17,15,27)$, $(31,19,21,16)$, and $(31,21,19,16)$. (We have $1/\infty = 0$, $1/0 = \infty$, and $\infty \mapsto \frac{a}{c} \bmod 47$.)

120. (a) The automorphisms are generated by $(i,j,k,u,v,w,x,y,z,U,V,W,X,Y,Z) \mapsto (i,j,k,Z,V,W,X,Y,U,z,v,w,x,y,u)$ or $(k,j,i,U,X,W,V,Y,Z,u,x,w,v,y,z)$. The cycles $u \,{-}\!{-}\, v \,{-}\!{-}\, i \,{-}\!{-}\, V \,{-}\!{-}\, Z \,{-}\!{-}\, Y \,{-}\!{-}\, W \,{-}\!{-}\, j \,{-}\!{-}\, w \,{-}\!{-}\, y \,{-}\!{-}\, z \,{-}\!{-}\, x \,{-}\!{-}\, k \,{-}\!{-}\, X \,{-}\!{-}\, U \,{-}\!{-}\, u$ and $Z \,{-}\!{-}\, V \,{-}\!{-}\, i \,{-}\!{-}\, v \,{-}\!{-}\, u \,{-}\!{-}\, y \,{-}\!{-}\, w \,{-}\!{-}\, j \,{-}\!{-}\, W \,{-}\!{-}\, Y \,{-}\!{-}\, U \,{-}\!{-}\, X \,{-}\!{-}\, k \,{-}\!{-}\, x \,{-}\!{-}\, z \,{-}\!{-}\, Z$ are quickly found by Algorithm H (just 44 nodes in the search tree).

(b) Since $G_0^{(t)}$ has a unique Hamiltonian path from $u^{(t)}$ to $U^{(t)}$, this construction uses it as a “gadget” to prevent any of the edges between Q_t and $G_0^{(t)}$ from appearing in any Hamiltonian cycle of G_t. The proof relies on the fact that $q_t \,{-}\!{-}\, l_t \,{-}\!{-}\, Q_t \,{-}\!{-}\, q_t \,{-}\!{-}\, r_t$.

(See H. Fleischner, *J. Graph Theory* **75** (2014), 167–177, Lemma 1. He goes on to define G_4 and G_5, thereby removing the degree-3 vertices y and Y in a similar way; those reductions introduce many more cycles, yet only one of them includes the edge $u \,{-}\!{-}\, U$. Another trick removes z and Z, in a graph $F = G_6$ that’s half of his tour-de-force!)

121. Let G be a graph with 25 vertices, one for each class of four cells $\{xy, y\bar{x}, \bar{x}\bar{y}, \bar{y}x\}$ that are rotationally equivalent, where $\bar{x} = 9 - x$. Adjacency is defined by giraffe moves; we must omit the self-loops from $\{23,37,76,62\}$ and $\{32,26,67,73\}$ to themselves. Furthermore, we remove one of the two edges between $\{22,27,77,72\}$ and $\{33,36,66,63\}$.

This 51-edge graph has 56 Hamiltonian cycles (found in just 33 Kμ); but the actual number is 100, because 44 of those cycles include the double edge. That yields 100 ways to cover a 10×10 board with symmetrical Hamiltonian cycles.

A cycle and its transpose are both counted. Hence there are exactly 50 essentially distinct solutions. (According to G. P. Jellis’s *Knight’s Tour Notes*, they were first discovered by T. W. Marlow, in 1998; see `www.mayhematics.com/t/pl.htm`.)

122. (a) For example, the triangles $\{\mathtt{b},\mathtt{e},\mathtt{f}\}$, $\{\mathtt{g},\mathtt{k},\mathtt{l}\}$, $\{\mathtt{d},\mathtt{i},\mathtt{j}\}$ must correspond somehow to the triangles $\{\mathtt{A},\mathtt{B},\mathtt{J}\}$, $\{\mathtt{F},\mathtt{G},\mathtt{H}\}$, $\{\mathtt{C},\mathtt{D},\mathtt{L}\}$. Hence the other vertices $\{\mathtt{a},\mathtt{c},\mathtt{h}\}$ and $\{\mathtt{E},\mathtt{I},\mathtt{K}\}$ must also correspond to each other in some order. The solution is

$$(\mathtt{a},\mathtt{b},\mathtt{c},\mathtt{d},\mathtt{e},\mathtt{f},\mathtt{g},\mathtt{h},\mathtt{i},\mathtt{j},\mathtt{k},\mathtt{l}) \leftrightarrow (\mathtt{I},\mathtt{J},\mathtt{K},\mathtt{H},\mathtt{A},\mathtt{B},\mathtt{L},\mathtt{E},\mathtt{F},\mathtt{G},\mathtt{C},\mathtt{D}).$$

[It’s unique, because this happens to be the “Frucht graph,” one of the smallest cubic graphs that has no automorphisms except the identity. See R. Frucht, *Canadian J. Math.* **1** (1949), 365–378. Halin graphs were introduced by R. Halin, *Combinatorial Mathematics and its Applications* (Oxford conference, 1969), 129–136.]

(b) For each nonleaf of T except the root, introduce the chord $i \,{-}\!{-}\, ((j+1) \bmod q)$ when its descendant leaves are $x_i \ldots x_j$. (The chords for `b`, `c`, `d`, `g` in the example are $0 \,{-}\!{-}\, 2$, $2 \,{-}\!{-}\, 5$, $5 \,{-}\!{-}\, 0$, $2 \,{-}\!{-}\, 4$, because $x_0 \ldots x_6 = \mathtt{efklhij}$.)

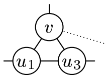
(c) Choose any region to be the root. The other regions and sides will form a tree T, when we ignore the adjacencies between sides of C, because the other adjacencies form no cycles. The children of each region in T are its adjacent regions and sides, except for the parent, in (say) clockwise order. For instance, if we choose root `I` in the example, the children of `I` might be (`J`, `K`, `H`); the children of `J` are (`A`, `B`); the children of `K` are (`L`, `E`); etc.; we could also have decided to let the children of `I` be (`K`, `H`, `J`) or (`H`, `J`, `K`). Or we could have chosen root `K`, with children (`E`, `I`, `L`) or (`I`, `L`, `E`) or (`L`, `E`, `I`); then the children of `I` would be (`H`, `J`), etc.

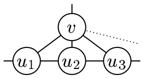
gadget
Fleischner
self-loops
transpose
Jellis
Knight’s Tour Notes
Marlow
Frucht graph
automorphisms
identity
historical notes
Halin

123. The answers for $4 \le n \le 16$ are 1, 1, 2, 2, 4, 6, 13, 22, 50, 106, 252, 589, 1475, computed by using definition (ii). See A. Howroyd, OEIS A346779 and A380362 (2025).

124. By induction on the number of vertices. The result is clear when tree T in (i) has depth 1. Otherwise some nonroot vertex $v \in T$ has $d \ge 2$ children $u_1 \ldots u_d$, all leaves.

Case 1: $d > 2$. Let H_0 be the Halin graph obtained by deleting leaf u_2. Then

H_0 has the subgraph 

where H has the subgraph 

.

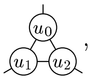
Since u_2 has degree 3, the Hamiltonian cycles of H have three possible forms:

$$u_2 \mathrel{-\!\!-} v \mathrel{-\!\!-} \alpha \mathrel{-\!\!-} u_1 \mathrel{-\!\!-} u_2, \quad u_2 \mathrel{-\!\!-} v \mathrel{-\!\!-} \alpha \mathrel{-\!\!-} u_3 \mathrel{-\!\!-} u_2, \quad u_2 \mathrel{-\!\!-} u_1 \mathrel{-\!\!-} \alpha \mathrel{-\!\!-} u_3 \mathrel{-\!\!-} u_2,$$

where the middle part is a Hamiltonian path of H_0. And such paths in H_0 arise from Hamiltonian cycles that respectively include the edges $v \mathrel{-\!\!-} u_1$, $v \mathrel{-\!\!-} u_3$, $u_1 \mathrel{-\!\!-} u_3$. So H has cycles that include/exclude $u_2 \mathrel{-\!\!-} u_1$, $u_2 \mathrel{-\!\!-} u_3$, $u_2 \mathrel{-\!\!-} v$. Furthermore, a Hamiltonian cycle of H_0 that excludes $u_1 \mathrel{-\!\!-} u_3$ must include $u_1 \mathrel{-\!\!-} v \mathrel{-\!\!-} u_3$; so we obtain Hamiltonian cycles in H that include/exclude $v \mathrel{-\!\!-} u_1$, $v \mathrel{-\!\!-} u_3$, and the edges from u_1 and u_3 to their anonymous neighbors. It's easy to include/exclude the other edges.

Case 2: $d = 2$. Now obtain H_0 by changing v to a leaf; in this case

H_0 has the subgraph

where H has the subgraph 

, letting $u_0 = v$.

By threefold symmetry, the Hamiltonian cycles of H that avoid edge $u_i \mathrel{-\!\!-} u_j$ correspond precisely to the Hamiltonian cycles of H_0 that avoid the "opposite" edge from v.

[Considerably more is also true; see Z. Skupień, *Contemporary Methods in Graph Theory* (Mannheim, 1990), 537–555. Uniform Hamiltonicity was introduced by C. A. Holzmann and F. Harary in *SIAM J. Applied Math.* **22** (1972), 187–193.]

125. $i_{97} \mathrel{-\!\!-} j_{97} = 58 \mathrel{-\!\!-} 54 = \pi_{78004}\pi_{78005} \mathrel{-\!\!-} \pi_{78006}\pi_{78007}$.

127. `GA` is impossible, because the arcs `AL` — `FL` — `GA` and `GA` — `SC` — `NC` are forced.

128. C, H, P, and U. (See exercise 103.)

129. (In the modified step H11, we can terminate the loop immediately if we encounter a vertex of degree 0 or 1.) The modified algorithm works surprisingly well: It wins convincingly on graphs A, G, and U (58 Mμ, 2763 Gμ, and 27 Gμ); it ties or does slightly better on graphs C, H, and T. It's slightly slower on graphs B, P, and Q; and it's more than 25% slower on graphs D, E, F, R, S.

130. In other words, if the present state of the computation can lead to a Hamiltonian cycle C, the current graph G' must have a Hamiltonian cycle C'. Indeed, that C' can be exhibited by replacing each subpath in C by the corresponding virtual edge of G'.

(Conversely, every Hamiltonian cycle of G' that actually uses every virtual edge corresponds to a unique Hamiltonian cycle C of G. There might, however, be other Hamiltonian cycles in G'. This graph G' was defined by W. Kocay in his paper of 1992, but he doesn't seem to have realized its full potential for pruning the search.)

One can, for example, discover whether or not G' has an articulation point by using Hopcroft and Tarjan's efficient depth-first algorithm for bicomponents (Algorithm 7.4.1.2B, or BOOK_COMPONENTS in *The Stanford GraphBase*).

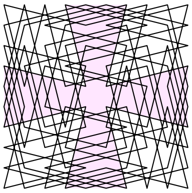
132. It would force a 4-cycle with two edges at the nearby corner.

Howroyd
OEIS
Skupień
Holzmann
Harary
historical notes
Kocay
pruning the search
articulation point
Hopcroft
Tarjan
depth-first
bicomponents

topological sorting
theoretical
practical
disjoint oriented cycles

133. For example, the fixed point $\alpha_1\alpha_2\alpha_3\alpha_4 = \bar\alpha_3\bar\alpha_2\bar\alpha_1\bar\alpha_4$ occurs if and only if $\alpha_1 = \bar\alpha_3$, $\alpha_2 = \bar\alpha_2$, and $\alpha_4 = \bar\alpha_4$ $(28\cdot 4\cdot 4$ cases). Summing over all eight fixed points yields the answer $(28^4 + 28 + 28^2 + 28 + 28^2 + 28\cdot 4\cdot 4 + 28^2 + 28\cdot 4\cdot 4)/8$. Similarly, without ‘a’, it's $(27^4 + 27 + 27^2 + 27 + 27^2 + 27\cdot 3\cdot 3 + 27^2 + 27\cdot 3\cdot 3)/8$.

888. Work in progress: This is the illustration promised on page 17.



150. (a) $v_1 \longrightarrow \cdots \longrightarrow v_n$ in G if and only if $s \longrightarrow v_1 \longrightarrow \cdots \longrightarrow v_n \longrightarrow t \longrightarrow s$ in $\widehat{G}$.

(b) Suppose $u_1 \longrightarrow \cdots \longrightarrow u_n$ and $v_1 \longrightarrow \cdots \longrightarrow v_n$ are Hamiltonian, with $u_j \neq v_j$ and j minimum. Then $u_j = v_k$ and $v_j = u_l$ for some $k > j$ and $l > j$. Consequently $u_l \longrightarrow v_{j+1} \cdots \longrightarrow v_k \longrightarrow u_{j+1} \longrightarrow \cdots \longrightarrow u_l$ is a cycle.

(c) True. We can assume that the vertices of G are labeled “topologically” (see Algorithm 2.2.3T) so that $v_j \longrightarrow v_k$ implies $j < k$. Algorithm B will choose $t \longrightarrow s \longrightarrow v_1$, since s and v_1 have in-degree 1. If $v_1 \longrightarrow \cdots \longrightarrow v_k$ have been chosen, for $k \geq 1$, then $\{v_1, \ldots, v_{k-1}\}$ are inner; hence v_{k+1} has in-degree 0 or 1. No backtracking is needed.

151. Construct $O(n^p)$ subgraphs of $\widehat{G}$ by removing one arc from each cycle in all possible ways. Then G has a Hamiltonian path if and only if at least one of those subgraphs is Hamiltonian. And each subgraph can be tested in $O(m+n)$ steps by exercise 150(c).

(Of course this result is purely theoretical, by no means practical!)

198. (a) There's one solution for every way to cover the vertices of G by disjoint oriented cycles of length ≥ 4. A cycle $u_0 \longrightarrow v_0 \longrightarrow u_1 \longrightarrow v_1 \longrightarrow u_2 \longrightarrow \cdots \longrightarrow v_{k-1} \longrightarrow u_0$ corresponds to choosing the options ‘u_0^- v_0 u_1^+’, ‘u_1^- v_1 u_2^+’, …, ‘u_{k-1}^- v_{k-1} u_0^+’.

(b) From the 332 options, Algorithm 7.2.2.1X needs about 180 Mμ to find 185868 solutions, of which $2\cdot 9862$ are the closed knight's tours (without removing symmetries).

199. Set up an exact cover problem as in exercise 198, where $n = 32$ and the vertices of the “first part” are the cells ij with $1 \leq i, j \leq 8$ and $i + j$ odd. Also add primary items $ij^\times$ for $i + j$ odd and $i > 2$. Each option now contains at least *six* items, not three: ‘u_1^- v_1 w_1^+ u_2^- v_2 w_2^+’ where $u_1 — v_1 — w_1$ and $u_2 — v_2 — w_2$, the six vertices are distinct, the i-coordinate of u_1 is less than the i-coordinate of u_2, and the j coordinates of (u_1, u_2), (v_1, v_2), (w_1, w_2) are equal. (The u's and w's belong to the “first part.” This option represents a pair of moves with matching columns.) Furthermore, append $ij^\times$ to each option for which $\{w_1, w_2\} = \{mj, ij\}$ or for which $\{u_1, u_2\} = \{mj, kj\}$ and $k \neq i$, where $m \in \{1, 2\}$. This trick forces the pairs of paths to “hook up” properly. For example, two of the options are ‘12^- 24 16^+ 52^- 44 56^+ $32^\times$ $72^\times$ $56^\times$’ and ‘41^- 22 43^+ 61^- 42 23^+ $43^\times$’. Exploit symmetry by removing options with $v_1 = 11$ and $w_1 = 32$.

That makes a total of 1998 options, and Algorithm 7.2.2.1X finds 383080 solutions in 14 Gμ. Each solution chooses 16 options, and a good one yields a cycle $(v_0 v_1 \ldots v_{63})$ in which the chosen options involve v_k^-, v_{k+1}, v_{k+2}^+, v_{k+32}^-, v_{k+33}, v_{k+34}^+ for $k = 0, 2, \ldots, 30$. Most solutions define short cyclic paths; but 5264 of them yield correct tours, such as the one shown.

1	32	57	30	3	12	55	16
58	29	2	11	56	15	52	13
33	64	31	4	35	54	17	50
28	59	34	41	10	51	14	53
7	40	63	36	5	22	49	18
60	27	6	9	42	19	46	21
39	8	25	62	37	44	23	48
26	61	38	43	24	47	20	45

200. Notice that we must have $a_{23} = 2$, $a_{28} = 17$, $a_{47} = 18$, $a_{67} = 50$, $a_{76} = 48$, $a_{88} = 49$. To find such a tour, we can begin by finding a knight's path of length 14 from step 2 to step 16 that doesn't interfere with 180° rotation, nor does it involve any of the "reserved" cells. All paths of length 14 are efficiently found by pasting together compatible paths of length 7. Useful paths also have the property that each vertex in the set U of cells available for steps $(18\,..\,30)$ and $(50\,..\,64)$ has degree ≥ 2 in the graph restricted to $U \cup I$, where $I = \{47, 52, 67, 32\}$ is the set of endpoints for future paths. The endpoints must also have degree ≥ 1 in that graph. A similar method finds 14-step paths for steps 18 through 32 and 50 through 64. One of the 46,596 solutions is shown.

1	30	9	6	3	16	61	24
10	7	2	15	62	25	4	17
31	64	29	8	5	38	23	60
28	11	14	63	26	59	18	37
13	32	27	58	51	36	39	22
54	57	12	45	42	21	50	19
33	46	55	52	35	48	43	40
56	53	34	47	44	41	20	49

201. Adapting the method in the previous exercise to paths of other lengths, we find that there are respectively (2, 47, 3217, 280244, 1205980, 259230, 41366) feasible solutions for the first (4, 9, 16, 25, 36, 49, 64) steps. The first full solution is shown. [*Brentano's Chess Monthly* **1**, 1 (May 1881), 36; **1**, 5 (September 1881), 248–249. See George P. Jelliss, *Mathematical Spectrum* **25** (1992), 16–20, for information about many similar "figured tours."]

1	4	9	16	25	36	49	64
8	15	24	3	10	17	26	37
5	2	7	14	35	50	63	48
56	13	34	23	18	11	38	27
33	6	55	12	51	40	47	62
54	57	22	19	46	43	28	39
21	32	59	52	41	30	61	44
58	53	20	31	60	45	42	29

250. Let the number be X_n, and let u, v, w be the "middle" vertices on the boundary. A Hamiltonian cycle on $S^{(3)}_{n+1}$ has the form $u \mathrel{—} \cdots \mathrel{—} v \mathrel{—} \cdots \mathrel{—} w \mathrel{—} \cdots \mathrel{—} u$, where the portions from u to v, v to w, and w to u are Hamiltonian *paths* from corner to corner of $S^{(3)}_n$. Consequently $X_{n+1} = Y_n^3$, where Y_n is the number of such paths.

Write uv for the corner between u and v. A Hamiltonian path from uw to vw has the form $uw \mathrel{—} \cdots \mathrel{—} u \mathrel{—} \cdots \mathrel{—} v \mathrel{—} \cdots \mathrel{—} vw$; and there are two cases, depending on whether w appears before u or after v. Thus $Y_{n+1} = Z_nY_nY_n + Y_nY_nZ_n$, where there are Z_n Hamiltonian paths from corner to corner in a graph that's like $S^{(3)}_n$ but with the third corner removed. Similarly, $Z_{n+1} = Z_nZ_nY_n + Y_nZ_nZ_n + [n\,{=}\,1]$.

We have $(X_1, Y_1, Z_1) = (1, 1, 1)$ and $(X_2, Y_2, Z_2) = (1, 2, 3)$. Hence, for $n \geq 3$, the formulas $X_n = 2^{3^{n-2}} 3^{3^1+3^2+\cdots+3^{n-3}}$, $Y_n = 3X_n$, $Z_n = \frac{3}{2}Y_n$ hold by induction.

We can in fact write $X_n = 8 \cdot 12^{(3^{n-2}-3)/2}$. [This problem was first solved by R. M. Bradley, *J. de Physique* **47** (1986), 9–14. See also A. M. Teguia and A. P. Godbole, *Australasian J. Combinatorics* **35** (2006), 181–192, who showed among other things that $S^{(3)}_n$ is *pancyclic*: It has cycles of every length, from 3 to $(3^n + 3)/2$.]

260. (a) True. The infinite rightward path that's traced by repetitions of the patterns will cross a vertical line once more when traveling to the right than when traveling to the left; it will cross a horizontal line equally often when going up as when going down.

(b) The total number of edges is $mn = v_1 + \cdots + v_{m-1} + h_1 + \cdots + h_n$. But the left side of this equation is even, while the right side is odd by (a).

(c) Frieze 5×4a in Fig. A–19 reduces to example (iv).

(d) The case $m = 1$ is trivial. When $m = 3$, the only possibilities are (i) and its cyclic shifts and/or left-right reversal; we consider them all to be equivalent (isomorphic), although the mirror reflection looks different. When $m = 5$ and $m = 7$, there are $3 + 10$ essentially distinct friezes, shown (except for (vi)) in Fig. A–19.

(e) Figure A–19 shows the $1 + 1 + 2 + 3$ solutions for $n = 5$, 6, 7, 8.

(We need a special convention when $n = 2$, because the "2-cycle" C_2 is a multigraph with two edges $0 \mathrel{—} 1$. We consider '⊓⊔⊓⊔' to be a 2×2 meander frieze. The 3×2a frieze reduces to it; the 2×4a frieze is a multiple of it.)

Jelliss
figured tours
Hamiltonian *paths*
Bradley
Teguia
Godbole
pancyclic
2-cycle

Historical notes
Jones
Chinese fretwork
fretwork
Coldstream

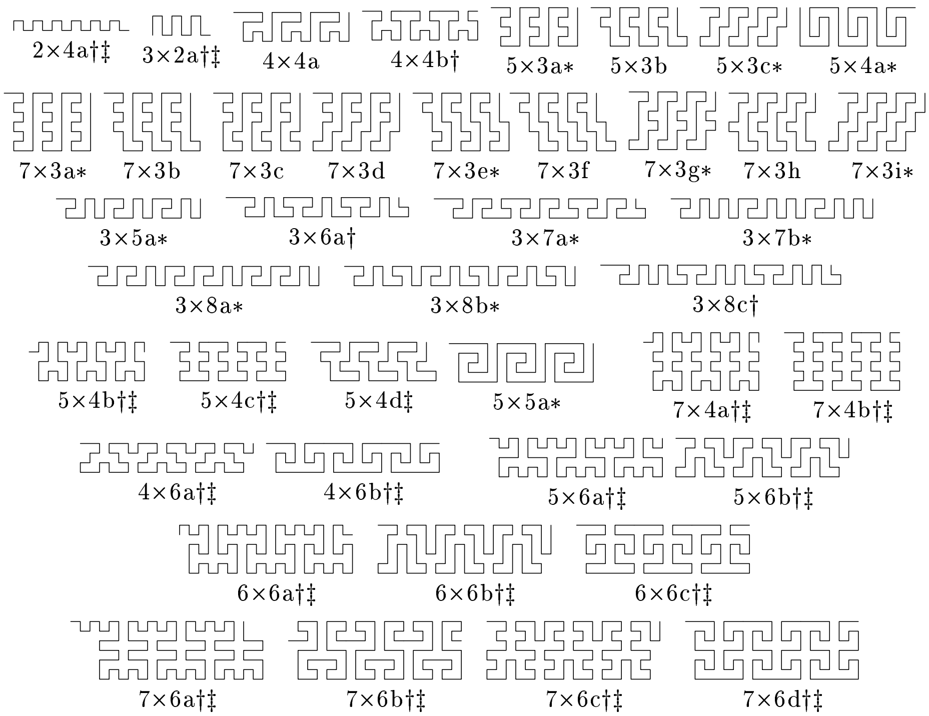


Fig. A–19. A gallery of meander friezes.
∗ = 180° symmetry; † = left-right symmetry; ‡ = top-bottom symmetry.

(f) The $4n$ automorphisms are generated by σ, τ, υ, where $ij\sigma = i((j+1) \bmod n)$, $ij\tau = i((-j) \bmod n)$, $ij\upsilon = (m-1-i)j$. Notice that $\sigma^n = \tau^2 = \upsilon^2 = (\tau\upsilon)^2 = 1$.

(g, h) The equivalence class sizes (with symmetric counts in parentheses) are:

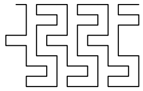
	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$
$m = 3$	1 (1)	1 (1)	1 (1)	1 (1)	2 (2)
$m = 4$	0 (0)	3 (2)	0 (0)	16 (8)	0 (0)
$m = 5$	3 (2)	12 (7)	43 (12)	154 (35)	534 (53)
$m = 6$	0 (0)	30 (5)	0 (0)	1152 (63)	0 (0)
$m = 7$	10 (4)	117 (27)	1216 (75)	12326 (383)	97969 (873)

(i) See (iii) and Fig. A–19. (Friezes 5×4d and 5×5a have only twofold symmetry.)

Historical notes: It's interesting to find instances of meander friezes in ancient artifacts from many cultures. For example, Chapter 4 of O. Jones's *The Grammar of Ornament* (1856) included (i) as a first example of a Greek design, and mentioned (ii) as a related pattern found in Chinese fretwork. J. N. Coldstream's book *Greek Geometric Pottery* (1968), which contains detailed information about the most important early discoveries of Greek vases from the Geometric Period, illustrates more than 40 specimens with the 3×3 frieze (i), and six with the 5×4 frieze (v). His Plate 7 shows three ancient vases with a symmetrical 7×4 frieze, which is related to 7×3e in Fig. A–19 as (v) is related to 5×3b. (And 5×3b, upside down, is in his Plate 13b.)

Dipylon Master meander friezes

The most elaborate meander frieze found so far in ancient sites is the magnificent 9×4 example shown here, due to the Dipylon Master and now in the collection of the National Archæological Museum in Athens. [See *Corpus Vasorum Antiquorum, Greece*, Fascicule 8 (Athens, 2002), Plates 102–105.]



261. (i) 00 — 01 — 02 — 03 — 13 — 23 — 22 — 12 — 11 — 21 — 20 — 10 — 00.
(ii) 00 — 01 — 11 — 10 — 20 — 21 — 22 — 23 — 13 — 12 — 02 — 03 — 00.
(iii) 00 — 01 — 02 — 03 — 13 — 12 — 11 — 21 — 22 — 23 — 20 — 10 — 00.
(iv) 00 — 01 — 11 — 21 — 22 — 12 — 02 — 03 — 13 — 23 — 20 — 10 — 00.
(v) 00 — 01 — 11 — 21 — 22 — 02 — 12 — 13 — 03 — 23 — 20 — 10 — 00.
(vi) 00 — 01 — 11 — 12 — 13 — 23 — 03 — 02 — 22 — 21 — 20 — 10 — 00.
(vii) 00 — 03 — 13 — 10 — 11 — 21 — 01 — 02 — 12 — 22 — 23 — 20 — 00.

Here (i) is in $P_3 \,\square\, P_4$; (ii) and (iii) are the meander friezes in exercise 260; (iv) is a multiple of the meander 3×2a; (vi) and (vii) are disjoint(!). These cycles have respectively 2, 4, 2, 8, 4, 2, 2 automorphisms; hence their equivalence classes contribute respectively 24, 12, 24, 6, 12, 24, 24 cycles to the total of 126 found by Algorithm H.

999. . . .

INDEX AND GLOSSARY

Florus, Lucius

Read the table that follows, my honest reader,
and it will soon guide you to hold the entire work in your mind.
— *Lucii Flori Bellorum Romanorum libri quattuor* (Vienna, 1511)

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.