

How many $n \times n$ British-style crossword grids are there?

Michael Keith

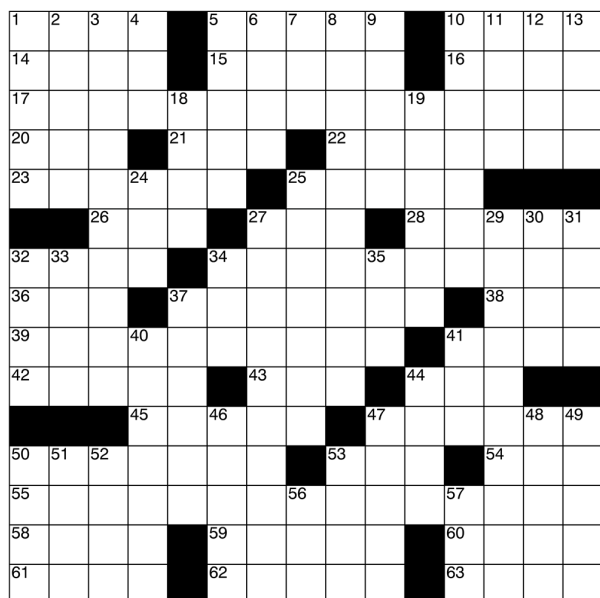
Gathering 4 Gardner 16, March 2026

Abstract

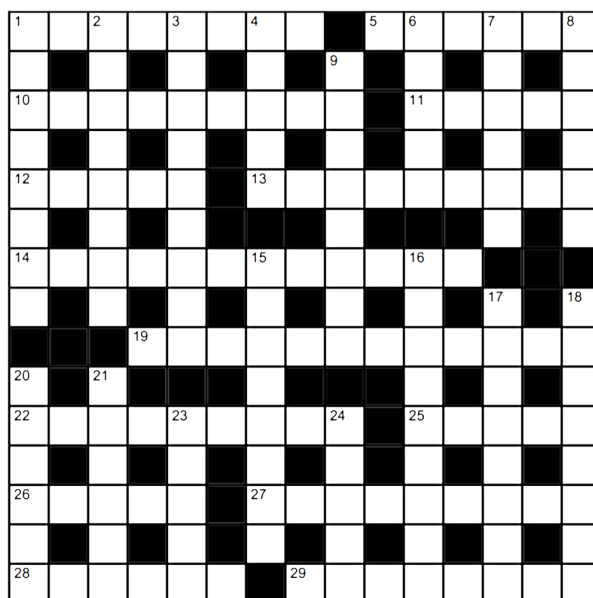
In a British-style cryptic crossword grid, the rules for placing white and black squares are quite different from an American-style puzzle. In this article we consider the problem of counting, and explicitly constructing, all distinct valid $n \times n$ British-style grids for odd $n \geq 5$. We'll also highlight special grids such as those with various symmetries, ones with the fewest/most words or letters, and grids having a nice distribution of word lengths.

Introduction

Typical American-style and British-style crossword grids are shown below, each example taken from the most canonical source: *The New York Times* and *The Times* (London), respectively.



American



British (cryptic)

The most common size for a crossword puzzle of either style is 15×15 , although 13×13 puzzles are also fairly common, such as the “Quick Cryptic” puzzle in *The Times*, which is not only smaller than a standard cryptic puzzle but also has easier clues. In this study we focus primarily on the small odd sizes: 5×5 , 7×7 , 9×9 , 11×11 , and 13×13 . Why not 15×15 ? We’ll answer that later.

We use the term *grid* for the pattern of white and black squares in a crossword, and seek to count and construct all *valid* grids for a given size, where a grid is valid if it satisfies a set of commonly-used rules. The rules for valid American grids and valid British grids are different (hence the different “look” of each), though several rules are shared between American grids and their

British cousins. Even though this article is about British grids exclusively, we'll describe the grid rules in a way that covers both types (American and British).

Rules for Valid Grids

Here are the rules we'll use to define valid grids. There's a set of five rules common to American and British puzzles, followed by two sets of style-specific rules.

Common Rules (American or British)

1. The grid is an $n \times n$ square, with n odd.
2. The pattern of white and black squares has 180° rotational symmetry.
3. Every word has at least 3 letters.
4. Every row and column on the outer edge has at least one white square.
5. All the white squares form a single 4-connected region.

American-only Rule

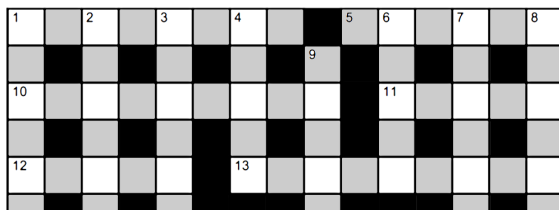
6. Every letter in every word must be checked.

British-only Rules

6. Each word of length k has exactly $\lceil k/2 \rceil$ checked letters.
7. Three or more adjacent unchecked letters are forbidden.
8. Two adjacent unchecked letters are allowed, but not at the start or end of a word.

We need to define what “checked” means. In any word of a puzzle (either Across or Down), a given letter is said to be *checked* if that letter is also part of a word in the other (perpendicular) direction. When a letter in a word intersects with another word in the other direction, we can check if we've filled in that letter correctly by figuring out the answer to the perpendicular word – hence the term “checked” for such a square.

In the partial British grid shown below, checked letters are white and unchecked letters are grey. Note that each complete word of length k has exactly $\lceil k/2 \rceil$ checked letters; for example, the 9-letter word at 10 Across has $\lceil 9/2 \rceil = 5$ checked letters.



It's important to mention that the rules listed above are a bit “squishy”, meaning that alternate versions of some rules are occasionally seen. In British puzzles, one or more of Rules 3, 6, and 7 are the most likely ones to be tweaked, typically in this fashion:

- 3* Minimum word length is larger than 3 (typically 4, could be higher).
- 6* Each word of length k has *either* $\lceil k/2 \rceil$ or $\lfloor k/2 \rfloor$ checked letters.
- 7* Two or more adjacent unchecked letters are forbidden (and there's no Rule 8).

Rules 3* and 7* are more restrictive than their normal counterparts, while 6* is less restrictive. It's tempting to use the more restrictive 3* and 7* to make the search for valid grids faster, but 3-letter words and pairs of unchecked squares appear often enough in real puzzles that we decided to go

with rules 3 and 7 as they are. Rule 6* only makes a difference when the word length is odd, and most puzzles seem to obey the more restrictive Rule 6, so we settled on the $\lceil k/2 \rceil$ version for this study.

Note that British Rule 6 is mostly responsible for the classic trellis-like look of a British grid, since it declares that half (rounded up) of the letters in each word are to be checked, leading to an arrangement where Across and Down words intersect at (roughly) every other letter. If all the words are in odd-numbered rows and columns, as in the example grid on the first page, the visual effect is even more pronounced.

Rules 4 and 5 taken together imply that no single row or column in the grid has all black squares. Rule 4 (whose purpose is to make sure that the 4-connected set of white squares touches all four edges of the $n \times n$ square) prohibits this on the edges, and an all-black row or column *not* on an edge splits the grid into two separate connected regions, in violation of Rule 5.

Now let's see some grids!

Results

Starting small, Figure 1 shows the 17 valid 5×5 grids. They are grouped in columns and colored to indicate which symmetries each grid has (in addition to the required 180° rotational symmetry). The five possibilities for these symmetries (only four of which occur in 5×5 s) are:

Name	Color	Meaning	Mult.
h	red	Horiz. reflection	2
r	green	90° rotation	2
t	blue	Transpose	2
A	tan	All of the above	1
N	white	None	4

Each of the five symmetry types (h, r, t, A, and N) has a multiplicity (last column), which says how many distinct grids are produced by applying all symmetries of the square to a grid with that symmetry type. For example, the green grid under the **r** in Figure 1 has symmetry type r because the only symmetry that leaves it unchanged is a 90° rotation. Type r has multiplicity 2, so a second grid is generated from this one by applying either one of the other symmetries (transpose

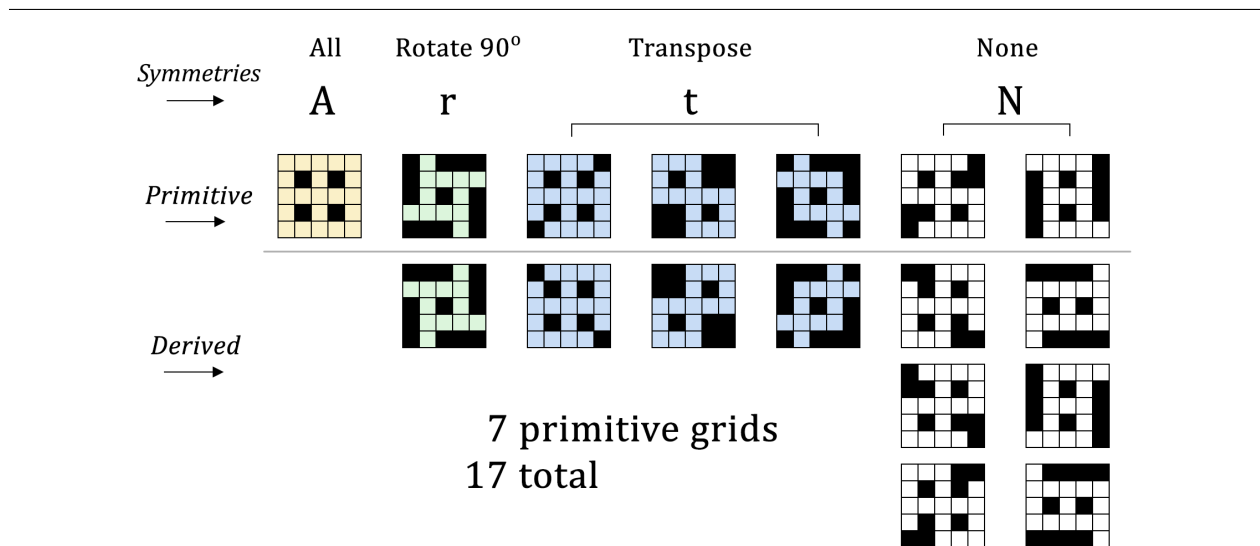
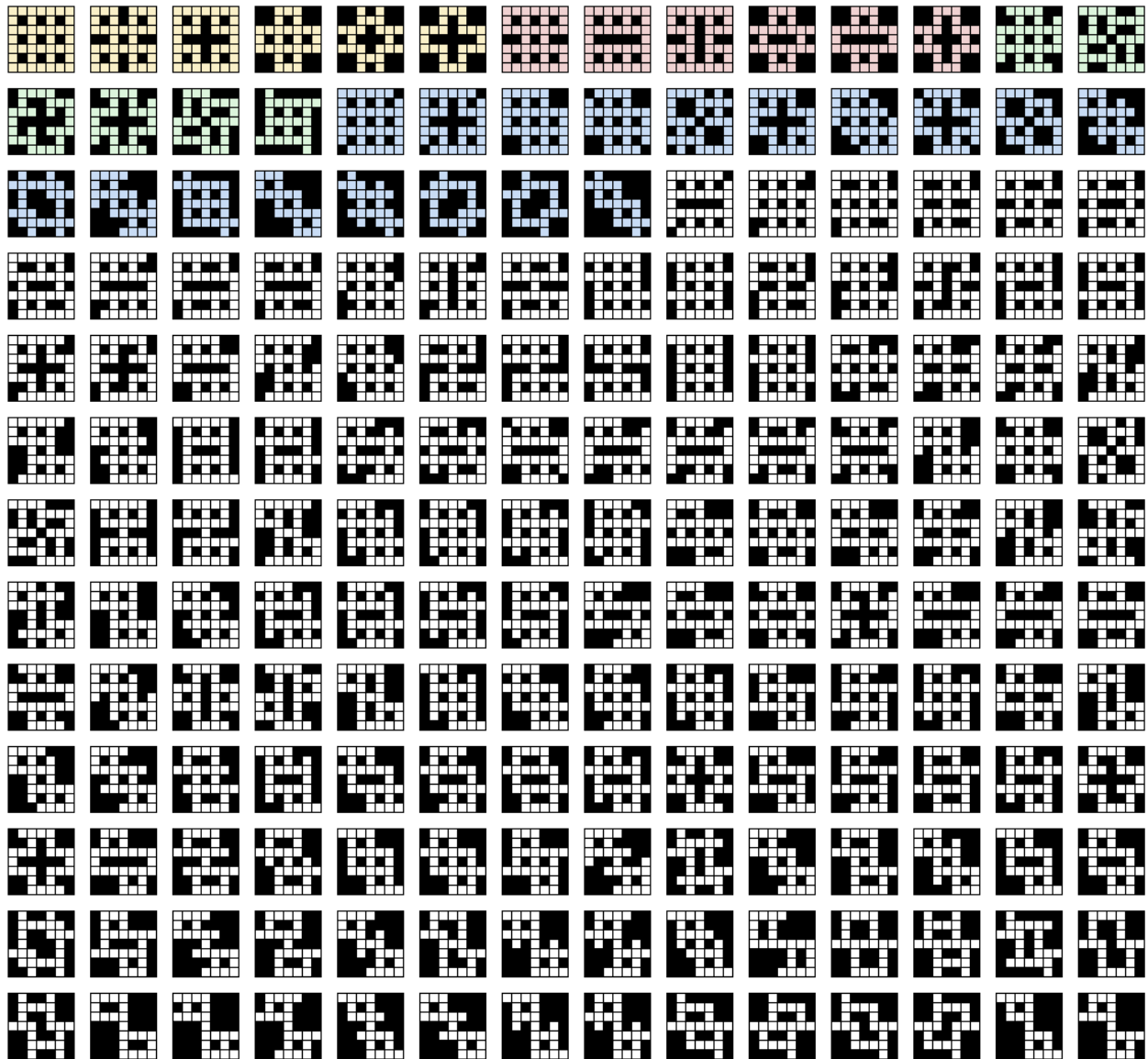


Figure 1: Valid 5×5 grids, classified by symmetry and primitiveness

or horizontal reflection). This *derived grid* is the green grid in the second row.

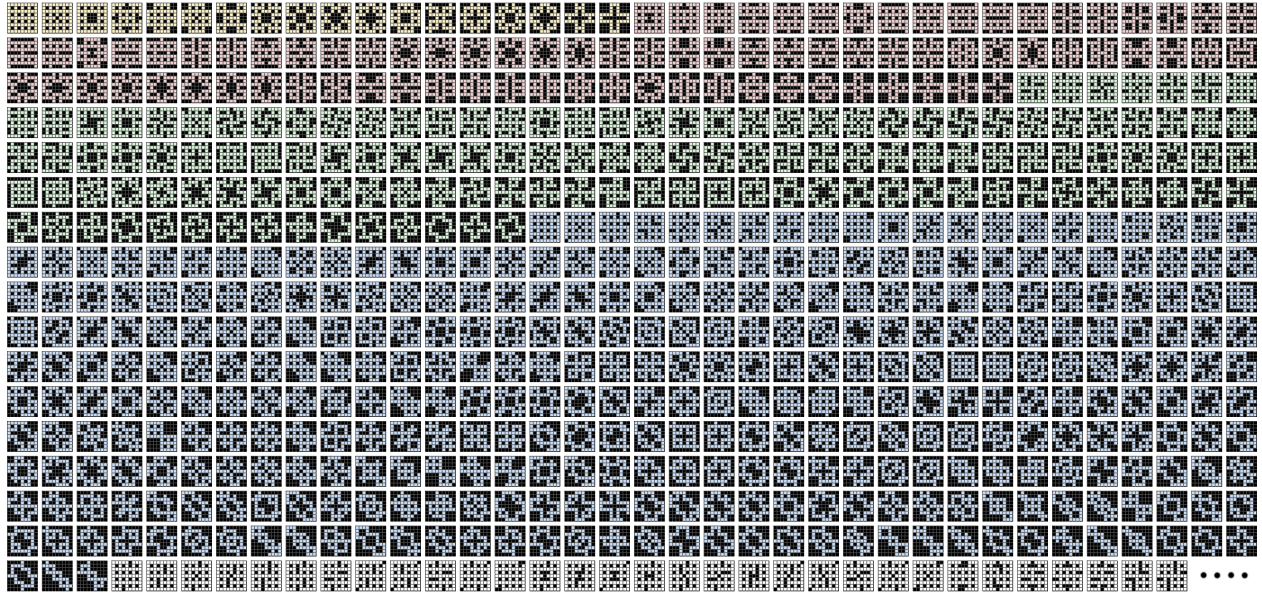
The information in Figure 1 can be condensed from the full list of 17 grids to the 7 *primitive* grids on the top row, each tagged with its symmetry type. Any grid can be chosen from each equivalence class (= column in Figure 1) to be the primitive grid. From now on we'll just list primitive grids, since we can easily generate the non-primitive solutions from them. This works nicely because all of the special grid properties we'll discuss are invariant under all symmetries of the square.

The 182 primitive 7×7 grids, which generate a total of 650 grids, happen to fit nicely in a 14×13 rectangle, as shown below. In lists like this, we order the grids of each symmetry type by increasing number of black squares. That there's a wide range of overall "brightness" in these grids leads to the idea of using all $n \times n$ grids, for some n , as a set of halftone patterns to create pictures. We'll explore that further in the final section.



There are too many 9×9 grids to list them all, but below are all the primitive grids of types A, h, r, and t, and the first few grids of type N. (You can display this PDF at 200% zoom, or more, to see these better.)

$\begin{matrix} 18 & \text{A} \\ 83 & \text{h} \\ 130 & \text{r} \end{matrix}$
 $\begin{matrix} 348 & \text{t} \\ 16754 & \text{N} \end{matrix}$
 \longrightarrow
 $\begin{matrix} 17,533 & \text{primitive grids} \\ 68,956 & \text{total grids} \end{matrix}$



The table below gives the number of primitive solutions for each symmetry type for sizes 5×5 to 13×13 , along with the total number of primitive grids and total number of all grids.

Symm (Mult)	5×5	7×7	9×9	11×11	13×13
A (1)	1	6	18	149	1,096
h (2)	0	6	83	5,372	283,977
r (2)	1	6	130	2,856	153,168
t (2)	3	18	348	9,984	588,555
N (4)	2	146	16,954	15,086,902	40,616,695,660
#Primitive	7	182	17,533	15,105,263	40,617,722,456
#Total	17	650	68,956	60,384,181	162,468,835,136
#Total Amer.	12	312	31,187	17,438,702	40,575,832,476

If A, h, r, t, N represent the number of primitive grids of each symmetry type, the two totals are given by

$$\begin{aligned} \# \text{Primitive} &= A + h + r + t + N \\ \# \text{Total} &= A + 2(h + r + t) + 4N \end{aligned}$$

For comparison, the bottom row of the table is the total number of American grids as computed by Jim Ferry in 2019 using the American version of our rules [1, 2]. By comparing the bottom two rows we see that there are more valid British grids than American for all n , with about four times as many for 13×13 . This may be surprising, since there are more rules for a British grid, but the main reason for this is that American Rule 6, which requires that all letters must be checked, is actually quite restrictive.

Programming Notes

Suspecting that this might be a time-consuming task, we attempted to design an efficient search program (in C++) from the start. For space reasons we'll just give a short overview of some of its features and optimizations:

1. Due to Rule 2 we only have to fill in the top half of the grid. We fill the half-grid one row at a time, moving from the middle row toward the top row, with a backtracking algorithm that at each step fills in the next row with only those patterns of white and black squares that are valid (given the rows already filled in).
2. Each row is a bit vector (1 = white, 0 = black) stored in a single integer variable. This allows us to do operations, such as checking if a candidate row is valid, in a bit-parallel fashion that's much faster than it would be if each grid square were a byte or integer.
3. The set of rows valid under rules 1-4 can be precomputed. Two lists are needed, one for the middle row (which must be a palindrome, by Rule 2) and one for the other rows. We postpone checking whether Rule 5 is satisfied until a grid has been completely constructed.
4. The most complex rules (6-8) can be tested all at once with a lookup table! To see how, suppose we've already filled in the middle row with value r_0 and the row above it with r_1 , and we're about to assign a candidate value to r_2 for the next row up (so they're in order r_0, r_1, r_2). We need to test whether r_0 and r_2 provide valid checks for r_1 . But a check is present if *either* r_0 or r_2 provides it, so we just have to know if $r_0 | r_2$ provides valid checks for r_1 , where $|$ is bitwise OR. We can concatenate $r_0 | r_2$ and r_1 to make a $2n$ -bit index into a 2^{2n} -sized one-bit lookup table which says if the inputs satisfy rules 6-8. For $n = 13$, for example, this table is 2^{26} bits = 2^{23} bytes ≈ 8.39 MB.
5. For a candidate row to be accepted, it must also be true that the set of partial columns satisfies all the rules. To check this, we transpose the binary grid and check if each group of 3 rows (2 on the edge) satisfies the rules. Note that if/when a single one of these tests fails, we can reject the candidate immediately. For an additional speedup we cache the transpose from the previous level of recursion so that just one bit on the left and right end of each row (in the transpose) has to be updated. As in item 4, a lookup table is also used to test the transpose rows, but it's a different table since we're testing incomplete rows (in which partial words at the start or end of a row must be ignored).
6. Separate code was written for each n , allowing a number of low-level optimizations to be used that depend on the size of the grid being a constant.

The execution times of our program for $n = 7, 9, 11$, and 13 are shown in the table below.

Size	1 CPU core	20 CPU cores
7×7	1.43 msec	0.07 msec
9×9	0.79 seconds	0.04 seconds
11×11	36.0 minutes	1.80 minutes
13×13	84.1 days	4.20 days

Comparing adjacent rows in the table shows that each increase in size increases the execution time by a factor of about 1000. Fortunately, this algorithm is “embarrassingly parallel” and can be doled out to multiple CPU cores or threads by giving each one a different value for the first (= middle) row. For 13×13 there are 83 different choices for the first row – more than enough for effective parallelization. The second column of the table above shows the execution time for the computing configuration we used: a set of four 4.5 GHz AMD Ryzen 7 PCs, each with 32 GB of RAM, with

the search running in parallel on 20 CPU cores.

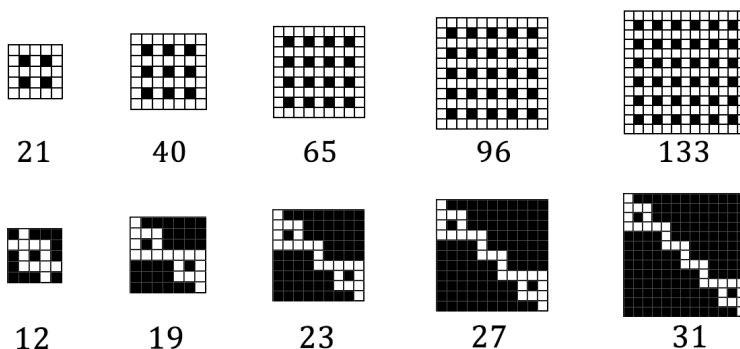
It's now clear why we didn't do 15×15 , because extrapolating the times in the third column of the table suggests that 15×15 would take many years. However, there are two positive factors that could come to the rescue:

1. For 15×15 it would be reasonable to use Rule 3* (with a minimum word length of 4) and Rule 7* (no adjacent pairs of unchecked letters), because many published 15×15 grids satisfy those rules. We estimate that this could provide a $\sim 50 \times$ speedup.
2. There's some chance that further optimization is possible. A further $2 \times$ to $5 \times$ speedup, combined with the rule change, *might* be able to bring the execution time into the feasible range (say, a few months).

Counting the number of 15×15 British grids is still an open problem.

Special Grids

Within all the primitive grids for each n are several subsets of special interest. For example, counting the number of white squares is interesting, since the range from "brightest" to "darkest" grids is quite large. Below are examples of grids with the most and fewest white squares for each n ; the number of white squares is shown below each grid.



The grids in the top row (most white squares) are unique for each n , but the dark grids on the bottom row are not. For example, there are 18 primitive solutions for 13×13 having 31 white squares. They all look somewhat similar to the one shown here, but the pattern in the upper left corner has several variations, and the zigzag of small words connecting them (all 3-letter words here) can also be various mixtures of 3- and 4-letter words.

Formulas for the maximum and minimum number of letters for each n are

$$L_{max} = O_{(n+1)/2} = \frac{(3n-1)(n+1)}{4} \quad (O_k = k^{\text{th}} \text{ octagonal number})$$

$$L_{min} = 2n + 5 \quad (\text{for } n \geq 7)$$

$L'_{max} = L_{max}/n^2$ (and similarly for L_{min}) represents the grid brightness normalized to $(0, 1)$. As $n \rightarrow \infty$, $L'_{max} \rightarrow \frac{3}{4}$ and $L'_{min} \rightarrow 0$, so for the idea of using grids as halftone patterns we get, in the limit, a brightness range from 0% to 75%. For small values of n it varies; e.g., for $n = 11$ the range is about 22% to 79%.

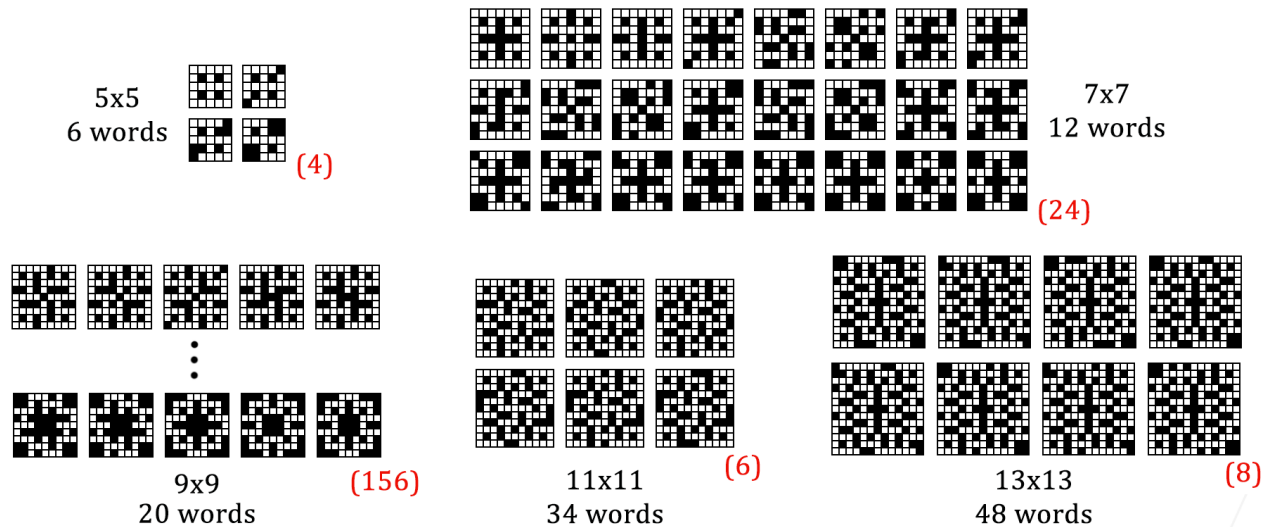
Even more interesting than counting white squares (letters) is counting words. Minimizing the number of words is the easiest, and there's a simple formula: $w_{min} = n - 1$. More than one grid achieves w_{min} for each n ; for $n = 5, 7, 9, 11 \dots$ there are 2, 3, 123, 305... such grids.

Here's one w_{min} grid for $n = 5, 7, 9,$ and 11 :



These all follow the same pattern, which can be continued, and illustrate that $w_{min} = n - 1$.

Determining the value of w_{max} for a given n is hard: currently, the only known way to compute it is to generate all the grids and find those with the maximum word count. Here are all primitive grids for $n = 5$ to 13 having w_{max} words (except for $n = 9$, where most are not shown):



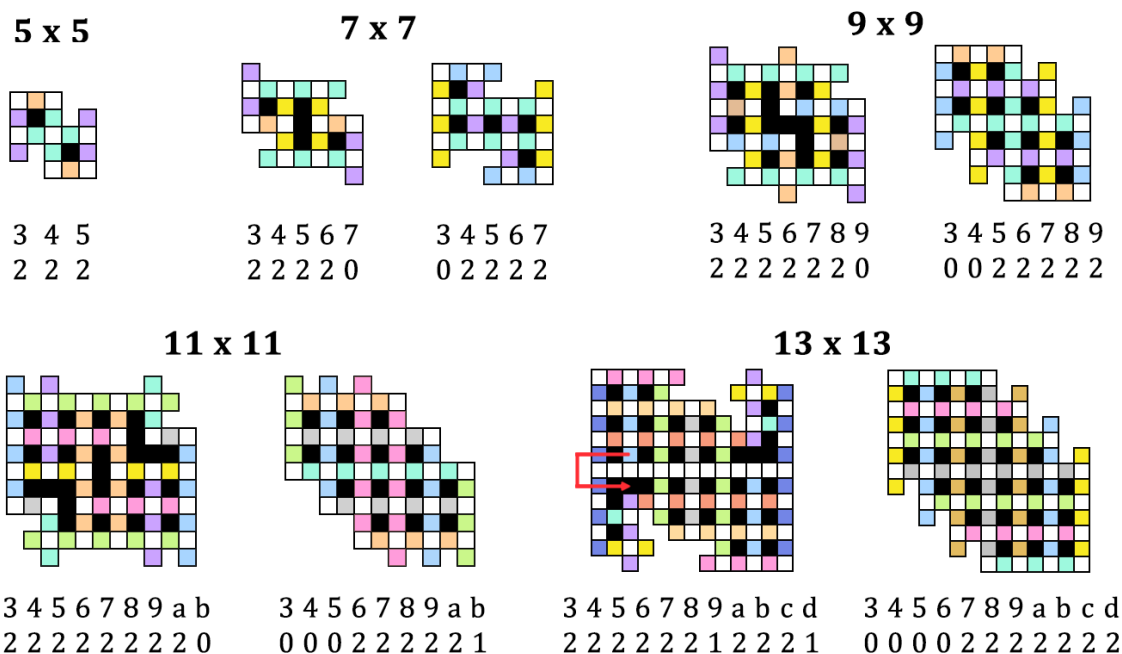
The sequence of w_{max} values begins 6, 12, 20, 34, 48..., which are all even (does this continue?). The red numbers indicate the number of grids with w_{max} words for each n . The eight 13×13 grids having the maximum possible 48 words are exceptionally nice, being the only 8 grids with this property in the whole set of 40,617,722,456 primitive 13×13 grids! The six 34-word 11×11 grids, from a total of 15,105,263 primitive grids, are also remarkable.

The last special property we'll explore concerns grids having certain pleasing combinations of word lengths. This feature is desirable to both setters and solvers of crosswords, for whom having a nice variety of lengths makes the puzzle more enjoyable. In general we'd like to maximize the number of distinct word lengths in the puzzle (ideally, having every word length present, from the minimum 3 to the maximum n) and also have an even(ish) distribution of the number of words of each length (ideally, with every word length appearing the same number of times).

We denote the word-length distribution of a puzzle by, for example:

3 4 5 6 7 8 9 a b
0 0 0 2 2 2 2 2 1

where the numbers in the top row are word lengths (with $a=10, b=11,$ etc.) and below each is the number of words of that length appearing in the puzzle. Here are some of the nicer word-length distributions we found:



Here we've employed an alternate way of drawing grids that's sometimes aesthetically useful: groups of black squares that touch the edge of the $n \times n$ square are not drawn. Also, every other letter of each word is colored to indicate its length, so that pairs of words of the same length (color) can be easily spotted.

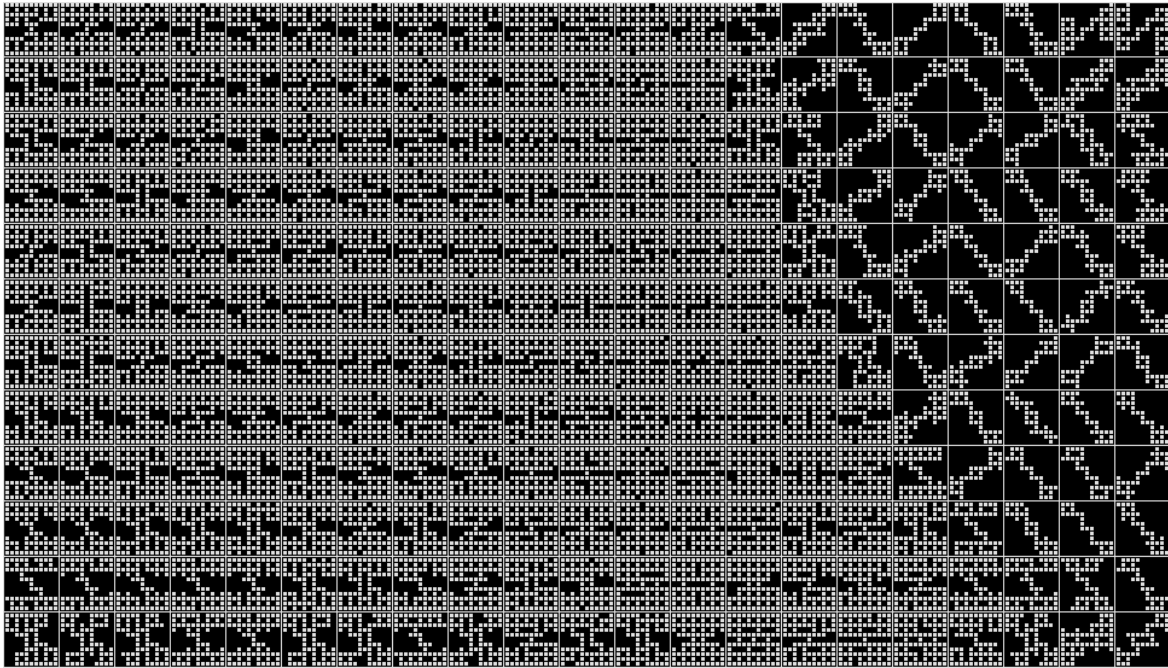
The 5×5 grid is the only "perfect" one: it has every possible word length and each length appears twice. None of our favorite grids for $n = 7, 9, 11$ contain all words lengths, but some of them have *almost* every length, with each length appearing the same number of times. The most beautiful of all of these must surely be the left grid of the 13×13 pair. It's the smallest grid after 5×5 containing every possible word length, and it is almost unique – the only other such grid is obtained by moving one square as indicated by the red arrow.

Does another grid exist, besides the 5×5 , with a perfect word-length distribution (all lengths are present and appear the same number of times)? If so, it must be 15×15 or larger.

Grids As Halftones

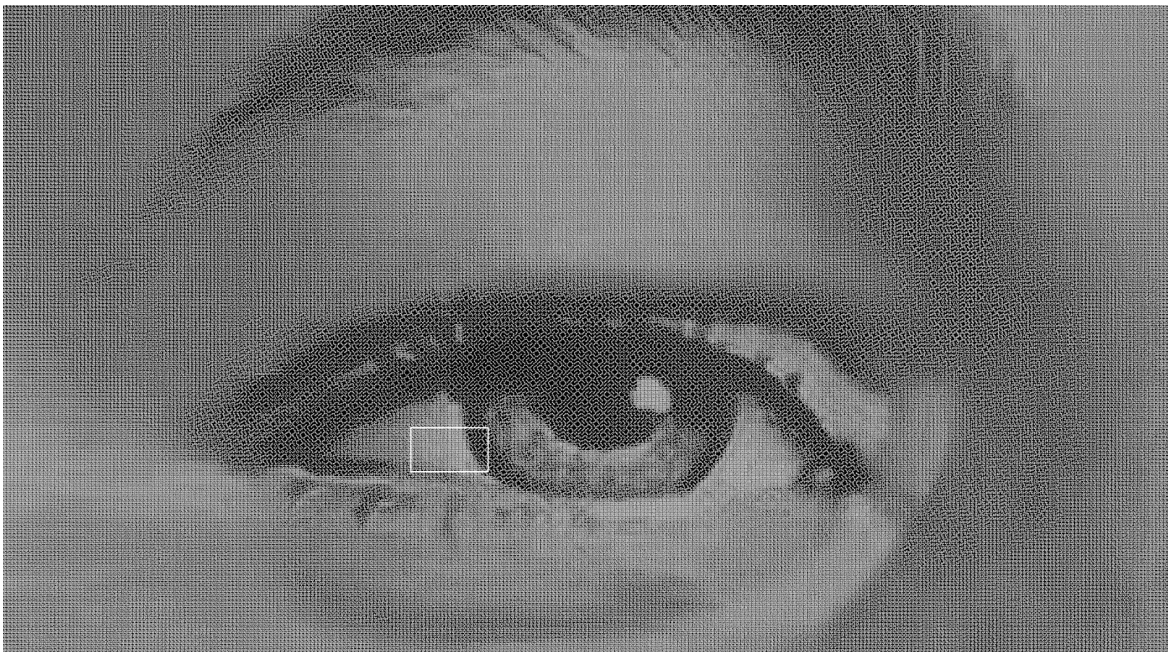
In this recreation the task is, for some n , to make a greyscale image using each primitive $n \times n$ grid exactly once. 9×9 is too small, giving a square image with only $\sqrt{17533} \approx 132$ pixels on a side, so we used 11×11 . The 11×11 grids span a brightness range of about 22% to 79%, meaning that the image will have a low-contrast look and perhaps some visible noise, but otherwise should be fairly accurate. To create the image we used a standard histogram-matching algorithm between the brightness distribution of the 11×11 grids and the target image, then randomly assigned a grid to each brightness level.

At the top of the next page is an extreme zoom into such an image, close enough to see the individual 11×11 grids. At the upper right is an area that's nearly black in the target image and therefore uses some of the darkest grids in the set. When displaying a dark grid such as these, we normally orient the diagonal strip of words to run from upper left to bottom right, but when making images out of grids it looks better if we randomly rotate each grid by 90° with probability $\frac{1}{2}$, to break up the monotonous diagonal pattern.



Amusingly, this random rotation is the same trick used in the classic Commodore 64 computer one-liner `10 PRINT CHR$(205.5 + RND(1)); : GOTO 10` to create a diagonal maze by randomly choosing between the two characters `\` and `/` [3]. Just turn your head 45° to see the partial maze in the upper right that results from randomly rotating each grid by either 0° or 90°.

Zooming out about 15× from the image above gives the image below. The white rectangle shows where the image above is located in this one.



One more zoom-out reveals the full image (see next page), a tribute to G4G16 featured speaker Donald Knuth. Finding the connection between this image and Dr. Knuth is left as an exercise for the reader.



References

- [1] Jim Ferry, *Crossword Counts* (2019), p. 21, available at <https://github.com/jimferry/crosswords-538/blob/master/Crossword%20Counts.pdf>.
- [2] A323839 in *The Online Encyclopedia of Integer Sequences*, <https://oeis.org/A323839>.
- [3] Nick Monfort et al., "10 PRINT CHR\$(205.5 + RND(1)); : GOTO 10", free PDF available at <https://10print.org>.
- [4] George Ho, *How Many Cryptic Crossword Grids Are There?* (2021), available at <https://www.georgeho.org/counting-cryptics/> (2021). **Remark:** supposedly enumerates all British grids for $n = 5, 7, 9$ using the same rules we used here, but the $n = 7$ and $n = 9$ counts are incorrect.