Crossword Counts

Solution to the January 18, 2019 edition of The Riddler

http://fivethirtyeight.com/features/how-many-crossword-puzzles-can-you-make/

Jim Ferry

February 7, 2019

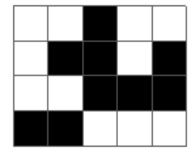
Overview

- The Mathematica package CrosswordCounts computes the number of possible crossword grids of various sizes
- These slides supplement the package by providing
  - A gallery of solutions for various cases
  - Examples of calls to CrosswordCounts used to produce the plots
  - A description of the dynamic programming technique that provides counts for problems too large to make explicit lists of solutions

- The Riddler's question: how many symmetric, 15 x 15 crossword grids are there with minimum word length 3?
- Answer:
  - 404,139,015,237,875 (tight case: grids must be fully 15 x 15)
  - 409,764,131,469,788 (loose case: grids may have all-black rows and/or columns surrounding a smaller, tight grid)

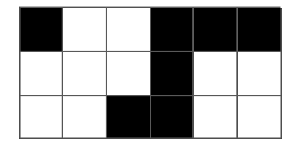
Definitions

- Grid: an m x n rectangle of black and white squares

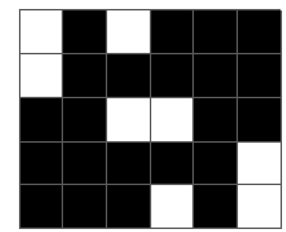


4 x 5 grid

3 x 6 grid with all word lengths ≥ 2

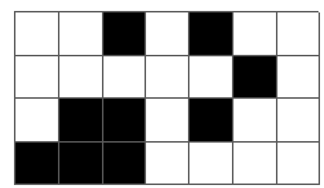


- Word: a run of horizontal or vertical white squares
  - Words extend until stopped by a black square or boundary
  - We will consider grids with all word lengths ≥ w, for various values of w
- Symmetric grid: a grid invariant under 180° rotation



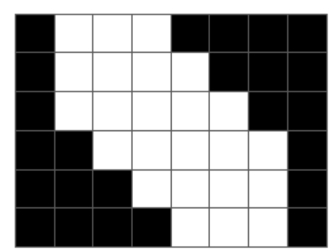
Symmetric 5 x 6 grid

- Connected grid: a grid whose white squares form one cluster
  - I.e., a white path exists between every pair of white squares (up/down/right/left moves only)



Connected 4 x 7 grid

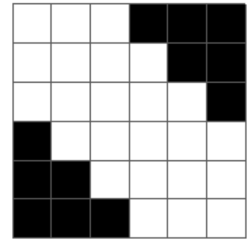
- <u>Valid grid</u>: symmetric, connected grid with all word lengths ≥ w
  - Main case of interest: w = 3



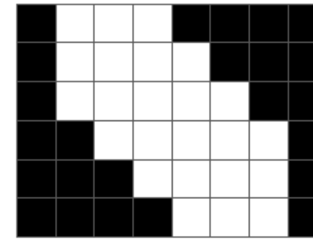
Valid 6 x 8 grid with all word lengths ≥ 3

Definitions

- Tight grid: grid with white squares touching every border
  - Loose case: non-tight grids okay
- Maximal grid: valid grid without "extra" black squares
  - See Slide 12 for precise defintion
- Symmetries: valid grids have 5 possible symmetry types
  - Color indicates symmetry types



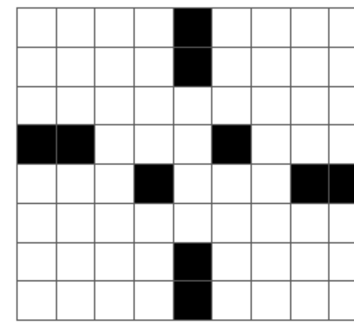
Tight, valid 6 x 6 grid with word lengths ≥ 3



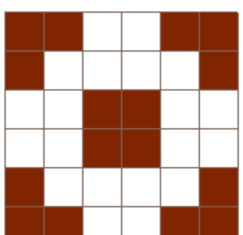
(Loose) valid 6 x 8 grid with word lengths ≥ 3

Maximal 8 x 9 grid with word lengths ≥ 3

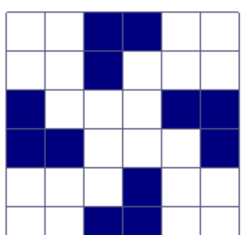
(maximal grids are always tight)



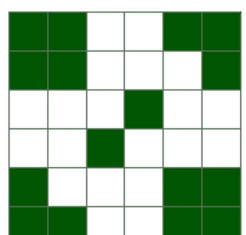
All



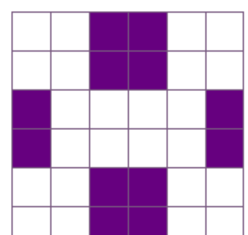
90° rotation



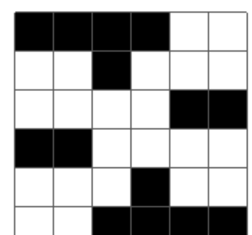
diagonals



horz/vert



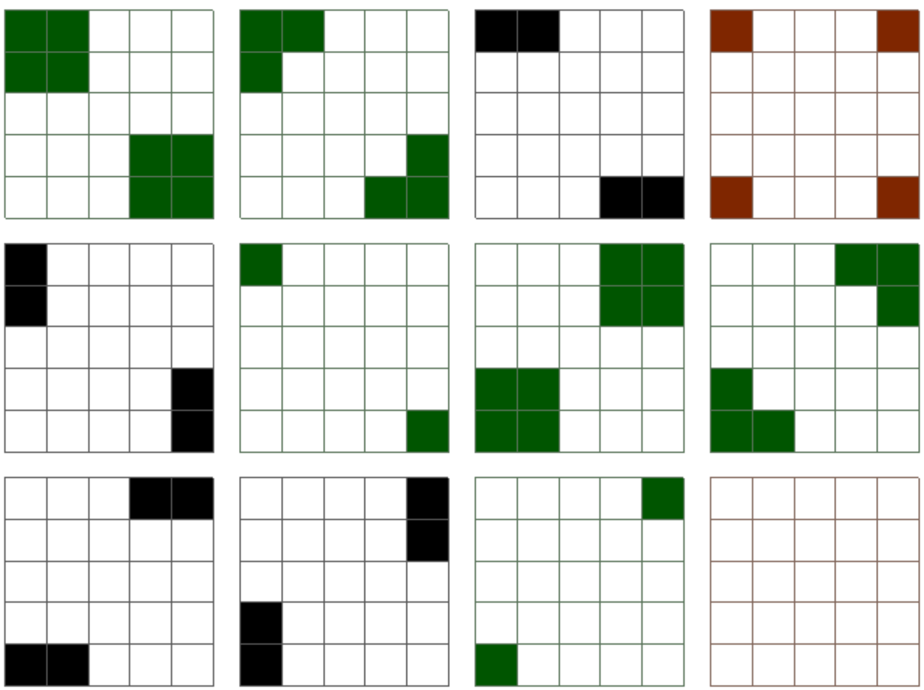
none (i.e., 180° rotation symmetry only)



Plotting All Grids vs. Grid Classes Only

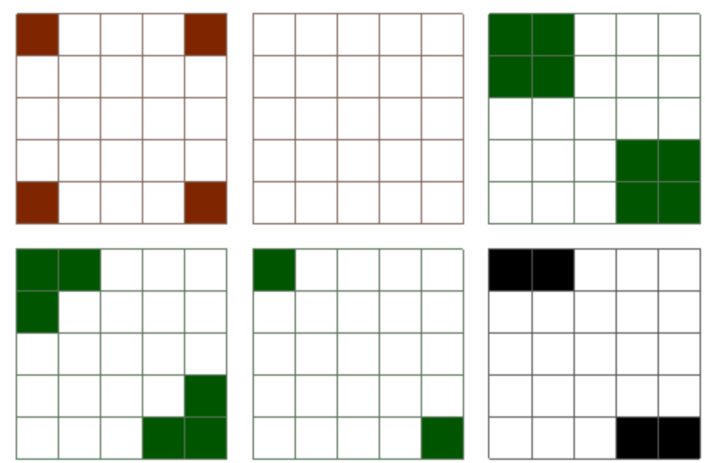
- Plots of all 12 valid 5 x 5 grids with word lengths ≥ 3
  - Note four “copies” of black grid

```
nValidGrids[5,3,1] = 12

plotGrids[allValidGrids[5,3,1]]
```



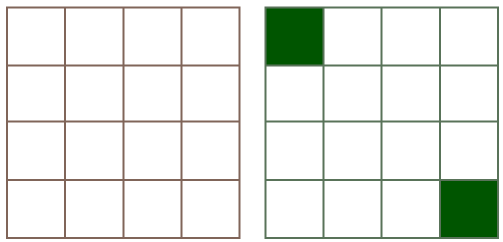
- Only need show examples from each “grid class”
  - Plots of all 6 valid 5 x 5 grid classes with word lengths ≥ 3
  - 2 red, 3 green, 1 black

```
nValidGridClasses[5,3,1] = 6

sizesValidGridClasses[5,3,1] =
{2,0,3,0,1}

plotGrids[allValidGridClasses[5,3,1]]
```

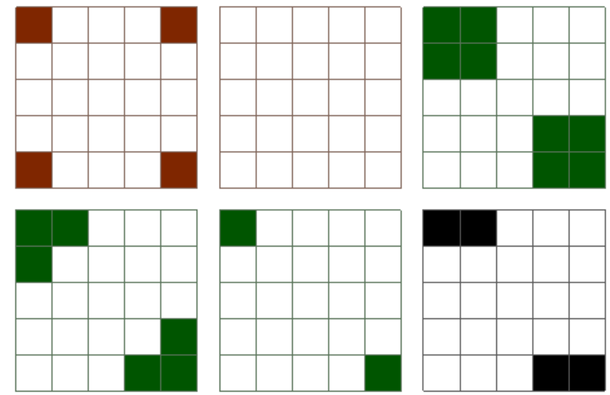


Valid Grids with w = 3

`plotGrids[allValidGridClasses[4,3,1]]`

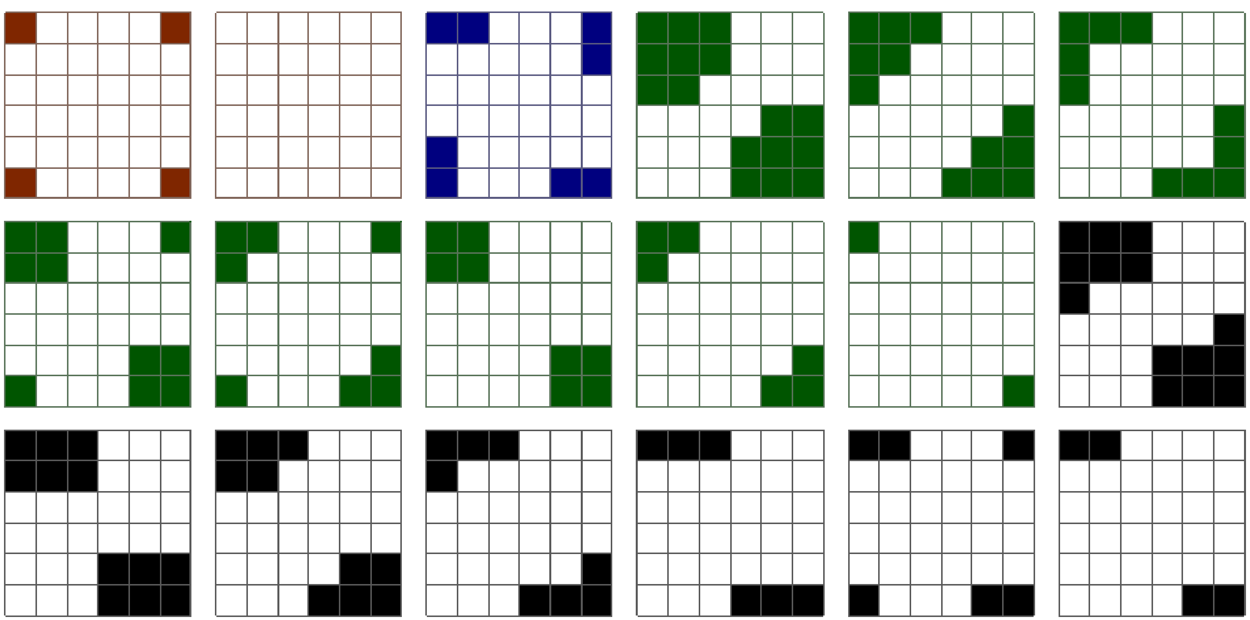


`plotGrids[allValidGridClasses[5,3,1]]`



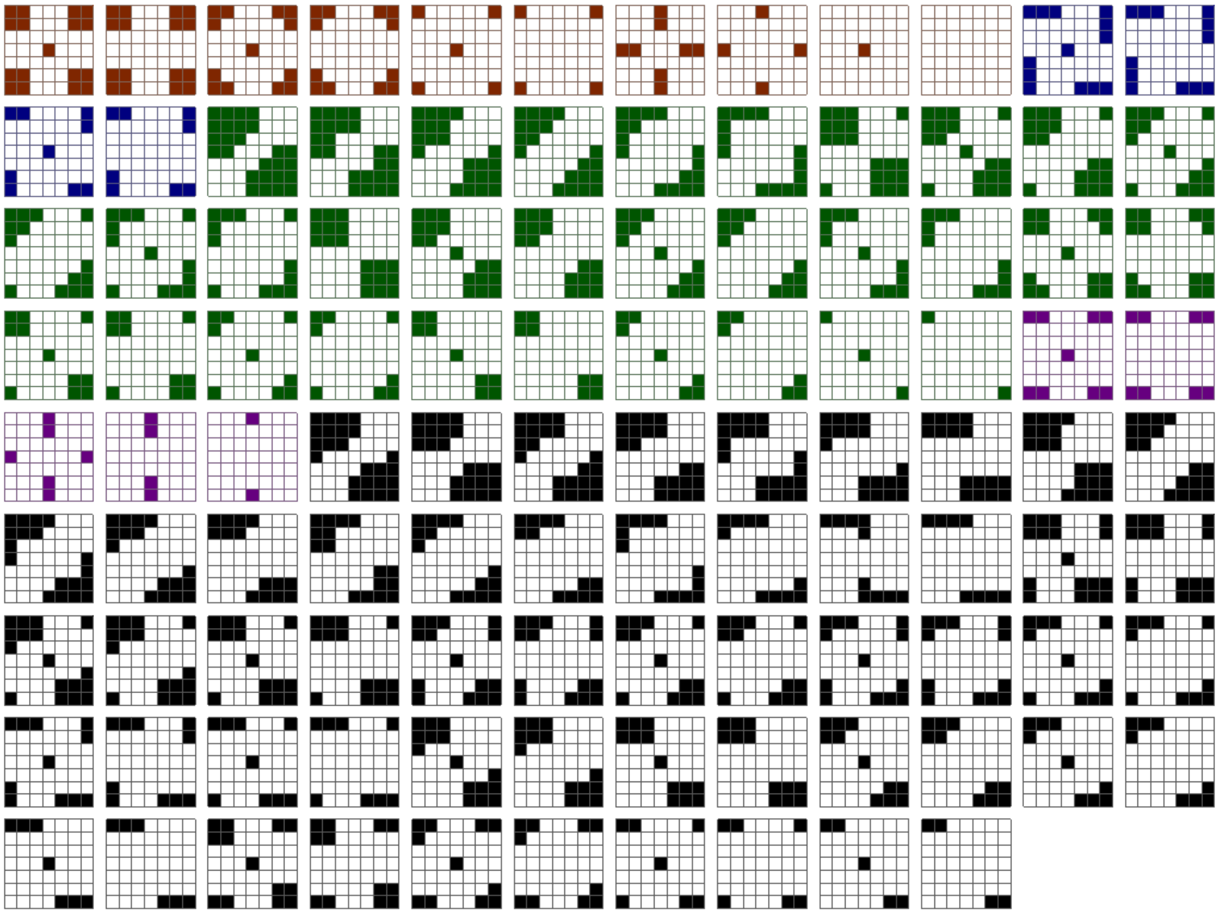
Only showing examples from each grid class

`plotGrids[allValidGridClasses[6,3,1]]`



Valid 7x7 Grids with w = 3

```
plotGrids[allValidGridClasses[7,3,1]]
```



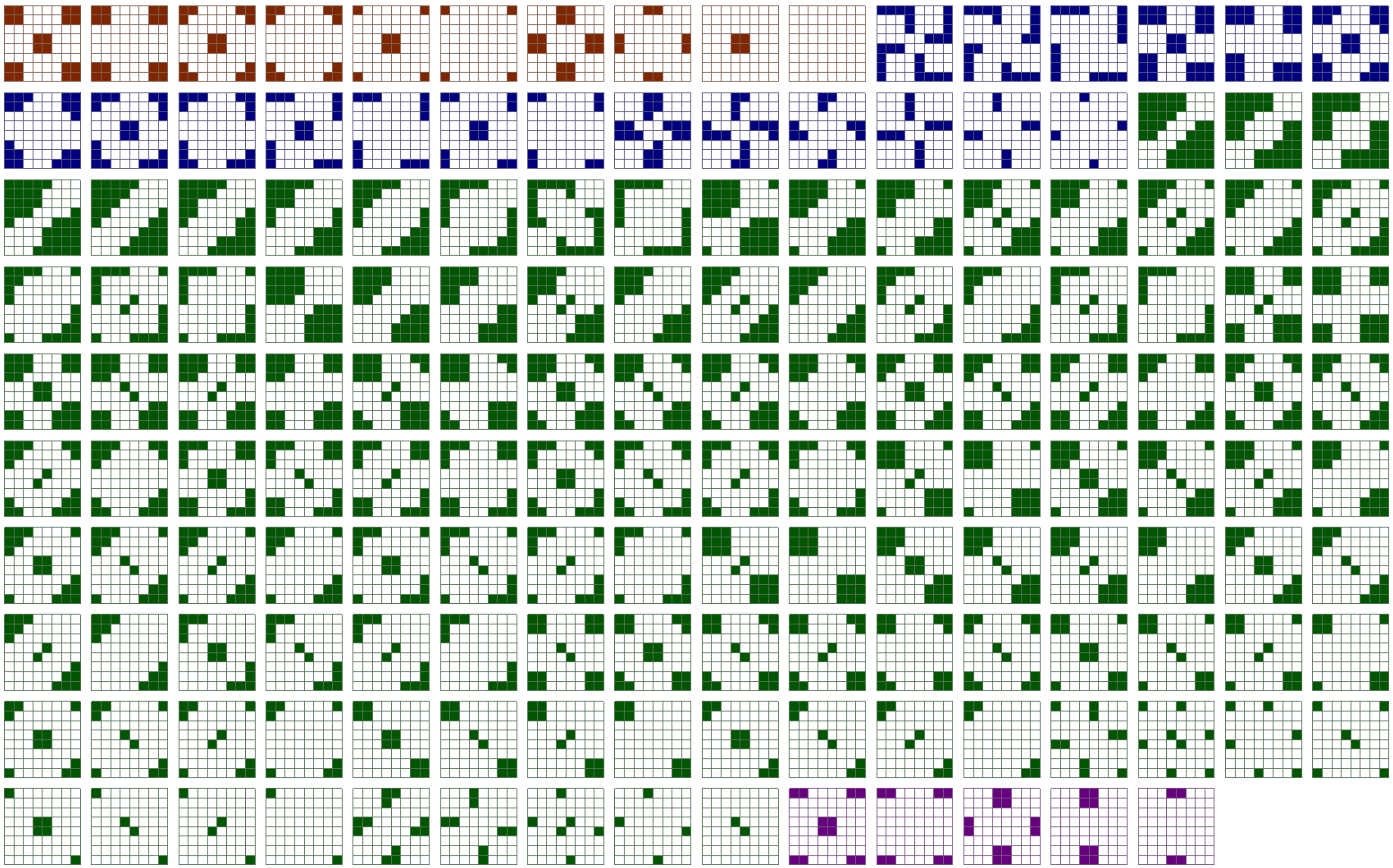
- Number of grids per class
  - Red: 1
  - Blue: 2
  - Green: 2
  - Purple: 2
  - Black: 4
- 106 grid classes
- 312 grids

```
nValidGridClasses[7,3,1] = 106      sizesValidGridClasses[7,3,1] = {10,4,32,5,55}

nValidGrids[7,3,1] = 312 (= 10*1 + 4*2 + 32*2 + 5*2 + 55*4)
```

Valid 8x8 Grids with w = 3

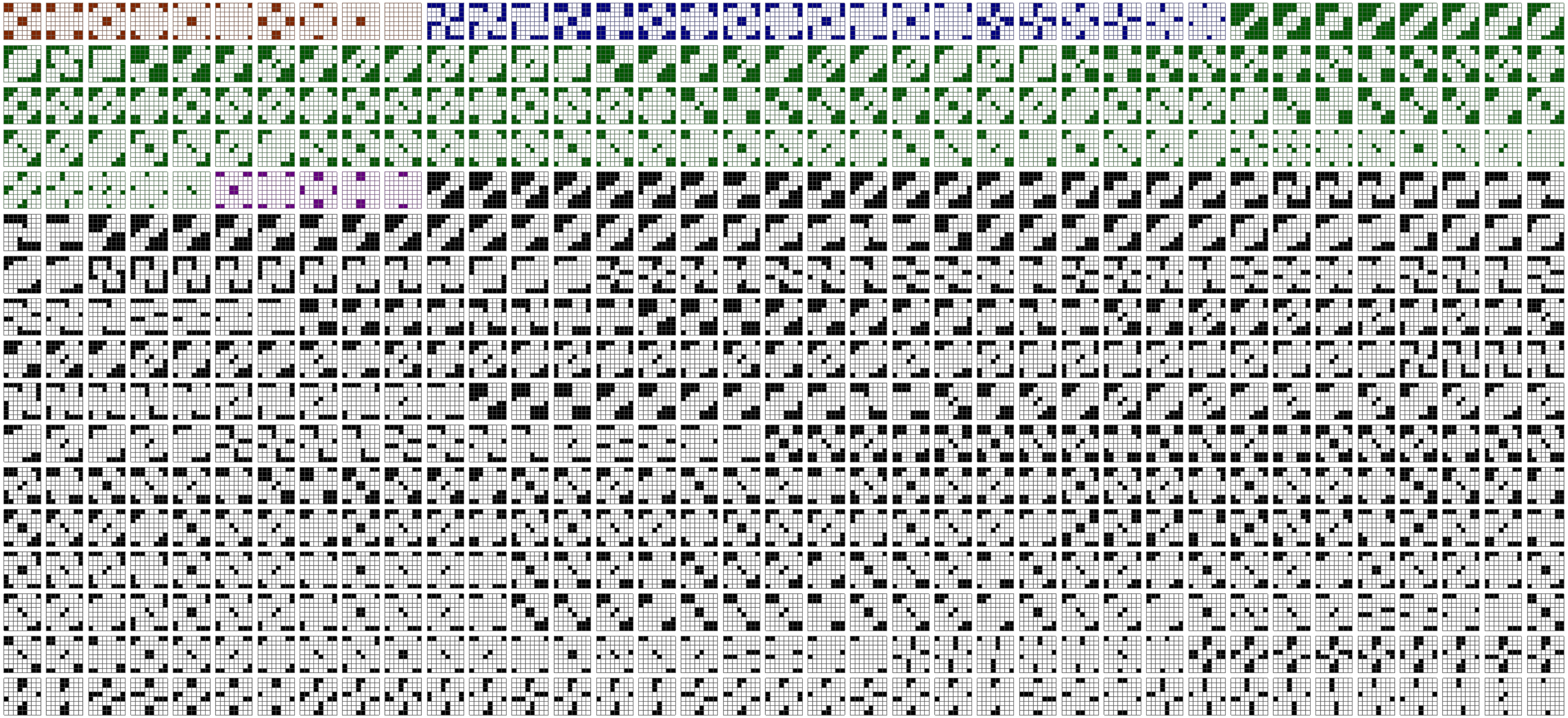
- 629 valid 8x8 grids with w = 3
  - Figure omits 471 black cases

```
sizesValidGridClasses[8,3,1] = {10,19,124,5,471}

plotGrids[allValidGridClasses[8,3,1][[;;4]]]
```



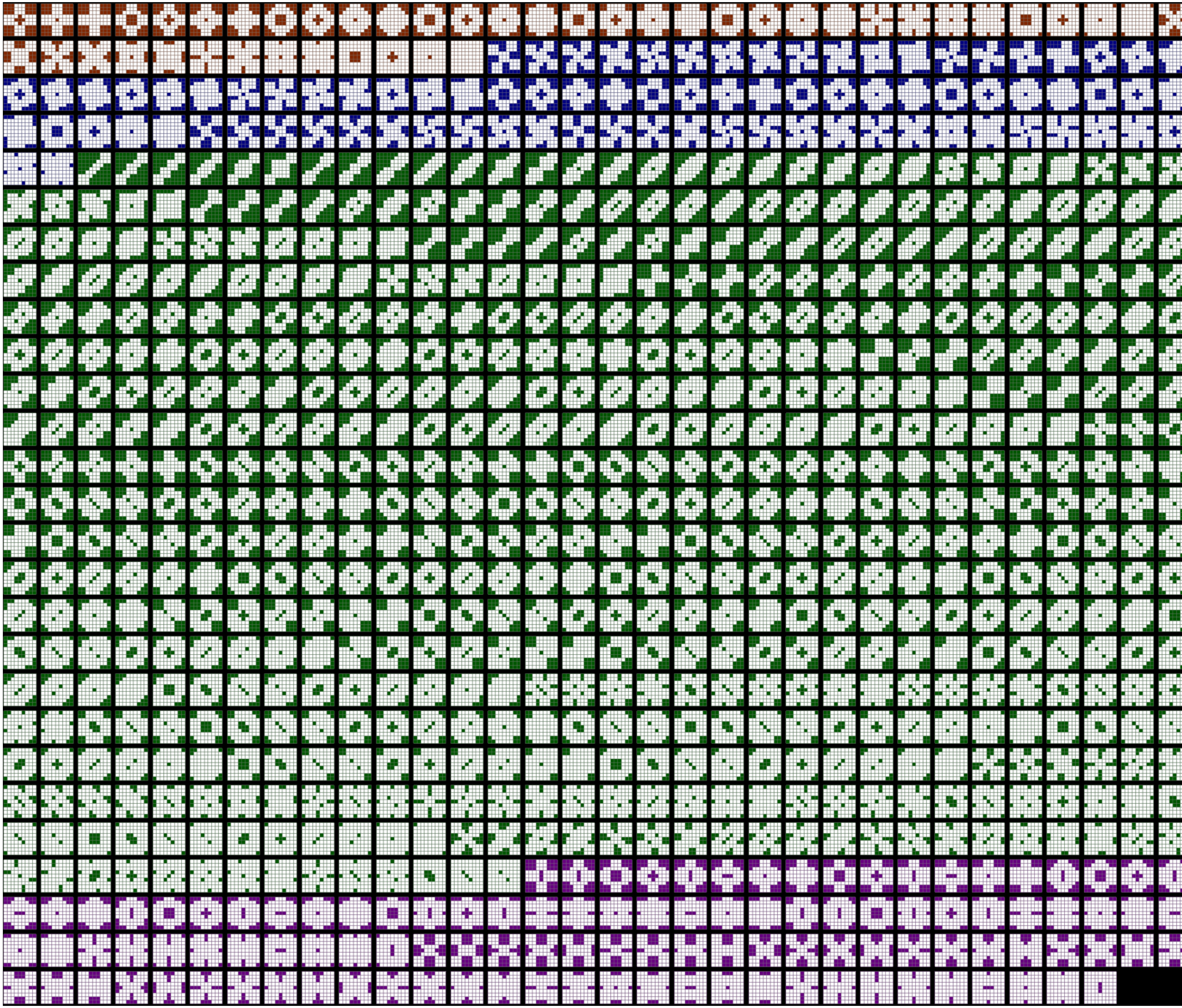
- All 629 valid 8x8 grids with w = 3

```
plotGrids[allValidGridClasses[8,3,1]]
```



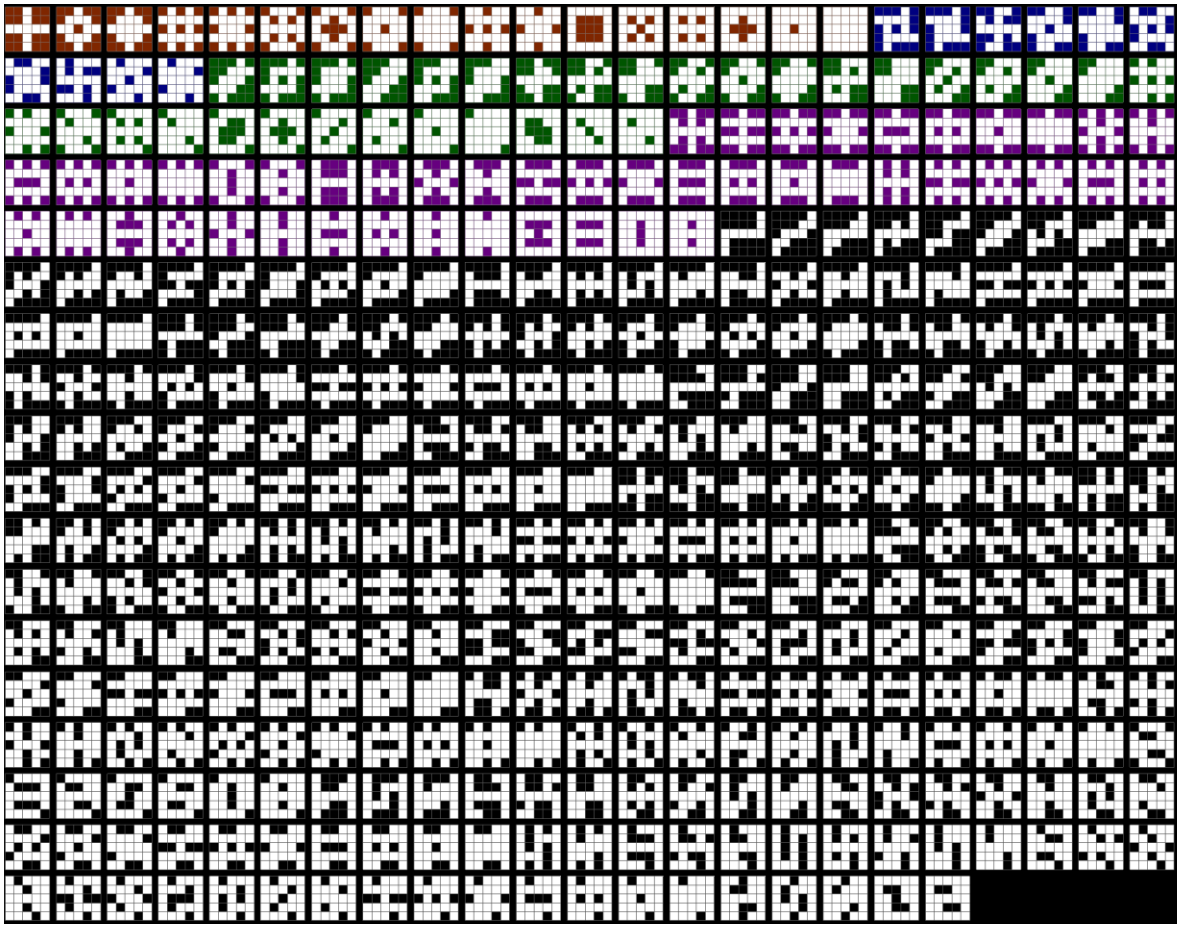
- Next slide
  - 8239 valid 9x9 grids with w = 3
  - Figure omits 7377 black cases

```
sizesValidGridClasses[9,3,1] = {45,85,620,112,7377}
plotGrids[allValidGridClasses[9,3,1][[;;4]]]
```

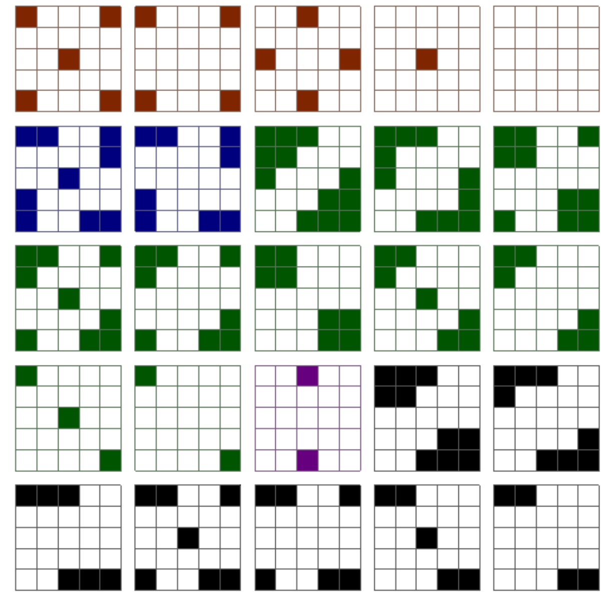


Valid 5x5 Grids for w = 1 to 4

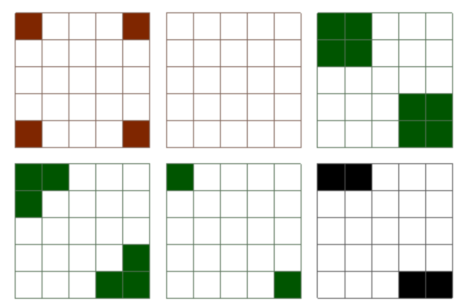
`plotGrids[allValidGridClasses[5,1,1]]`



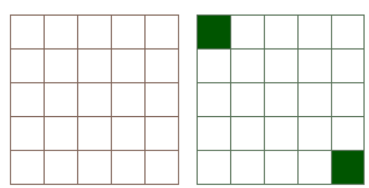
`plotGrids[allValidGridClasses[5,2,1]]`



`plotGrids[allValidGridClasses[5,3,1]]`



`plotGrids[allValidGridClasses[5,4,1]]`



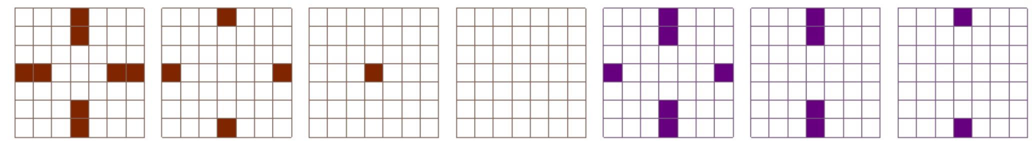
Maximal Grids

- Let V(m,n,w) be all valid m x n grids with word lengths ≥ w
  - Most such grids have too many black squares to be good crosswords
- Define idea of a *maximal grid* based on a *partial order* ≤
  - For any two m x n grids G and H, define H ≤ G when
    - Every white square in H is a white square in G
    - Any two white squares in different words in H are in different words in G
  - Let M(m,n,w) be the maximal elements of V(m,n,w)
    - I.e., those H in V(m,n,w) for which no G in V(m,n,w) satisfies H ≤ G
- The maximal grids M(m,n,w)
  - Are always tight (even if V(m,n,w) includes non-tight grids)
  - Are a small subset of V(m,n,w)
  - Look like actual crossword puzzles

Maximal Grids with w = 3

- Valid 7 x 7 grids
  - Loose: 398
  - Tight: 312
  - Maximal: 10

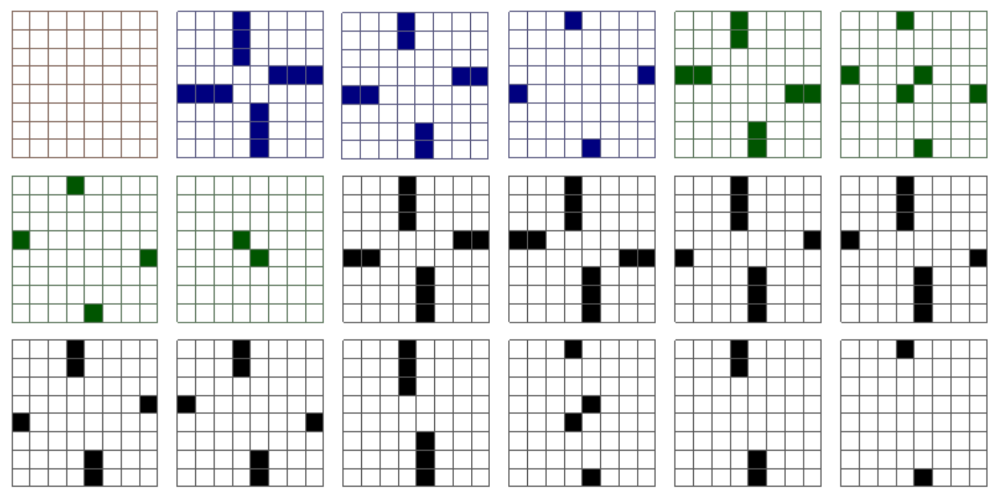
```
plotGrids[allMaximalGridClasses[7,3],30,7]
```



```
nValidGrids[7,3,0] =  398    sizesMaximalGridClasses[7,3] =
nValidGrids[7,3,1] =  312                           {4,0,0,3,0}
nMaximalGrids[7,3] =   10  (= 4*1 + 3*2)
```

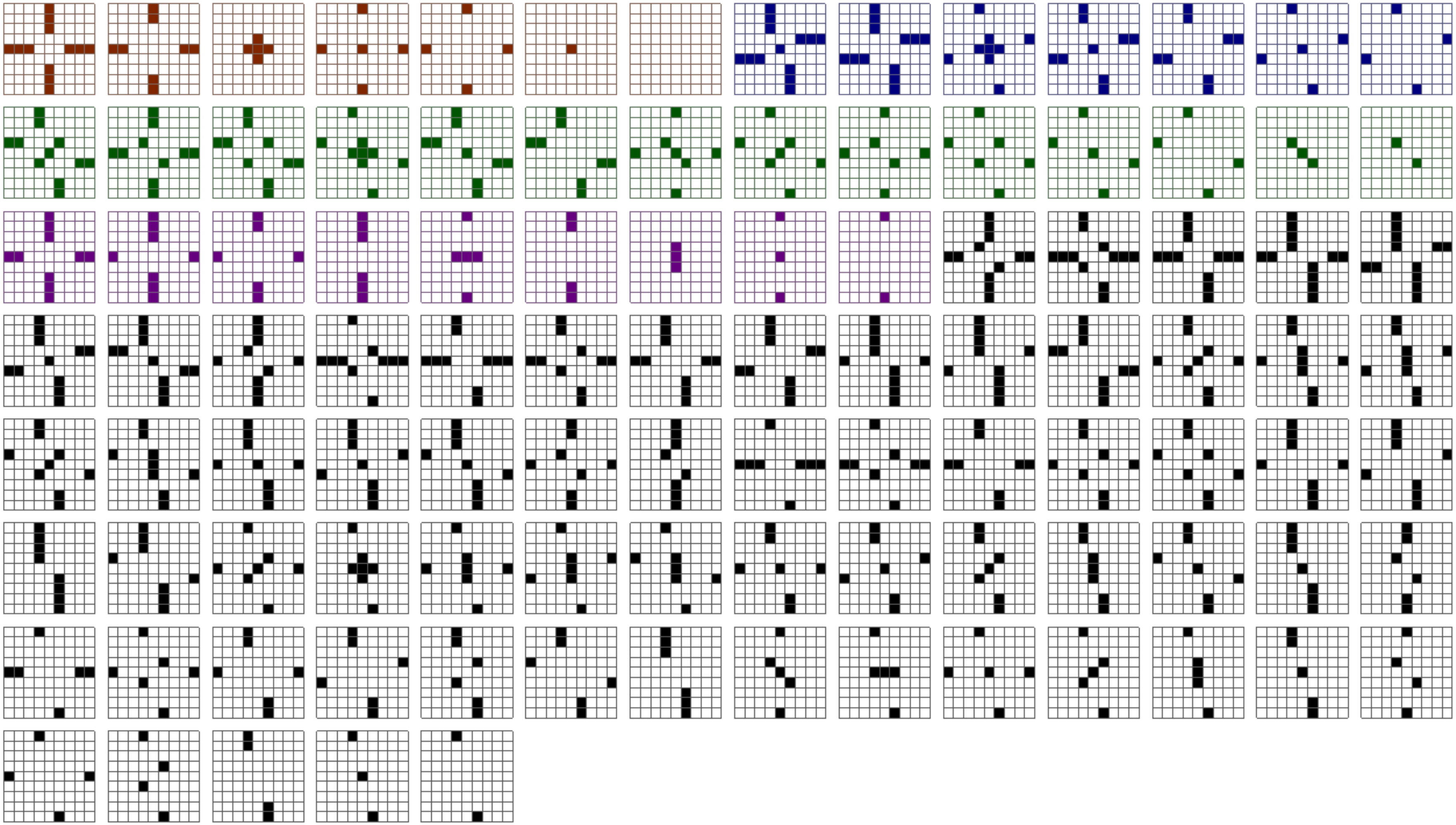
- Valid 8 x 8 grids
  - Loose: 2638
  - Tight: 2190
  - Maximal: 55

```
plotGrids[allMaximalGridClasses[8,3]]
```

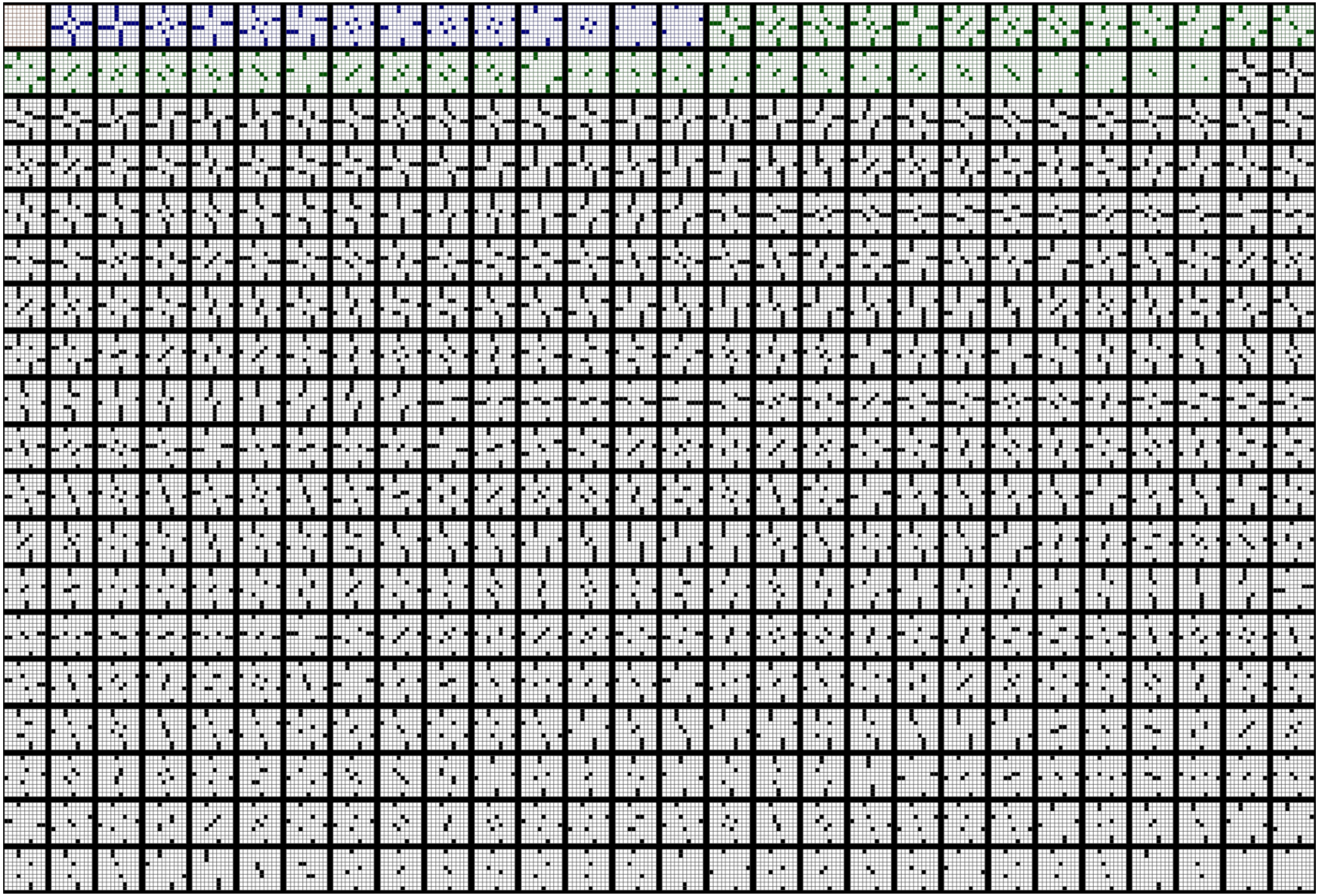


```
nValidGrids[8,3,0] = 2638    sizesMaximalGridClasses[8,3] =
nValidGrids[8,3,1] = 2190                          {1,3,4,0,10}
nMaximalGrids[8,3] =   55  (= 1*1 + 3*2 + 4*2 + 10*4)
```

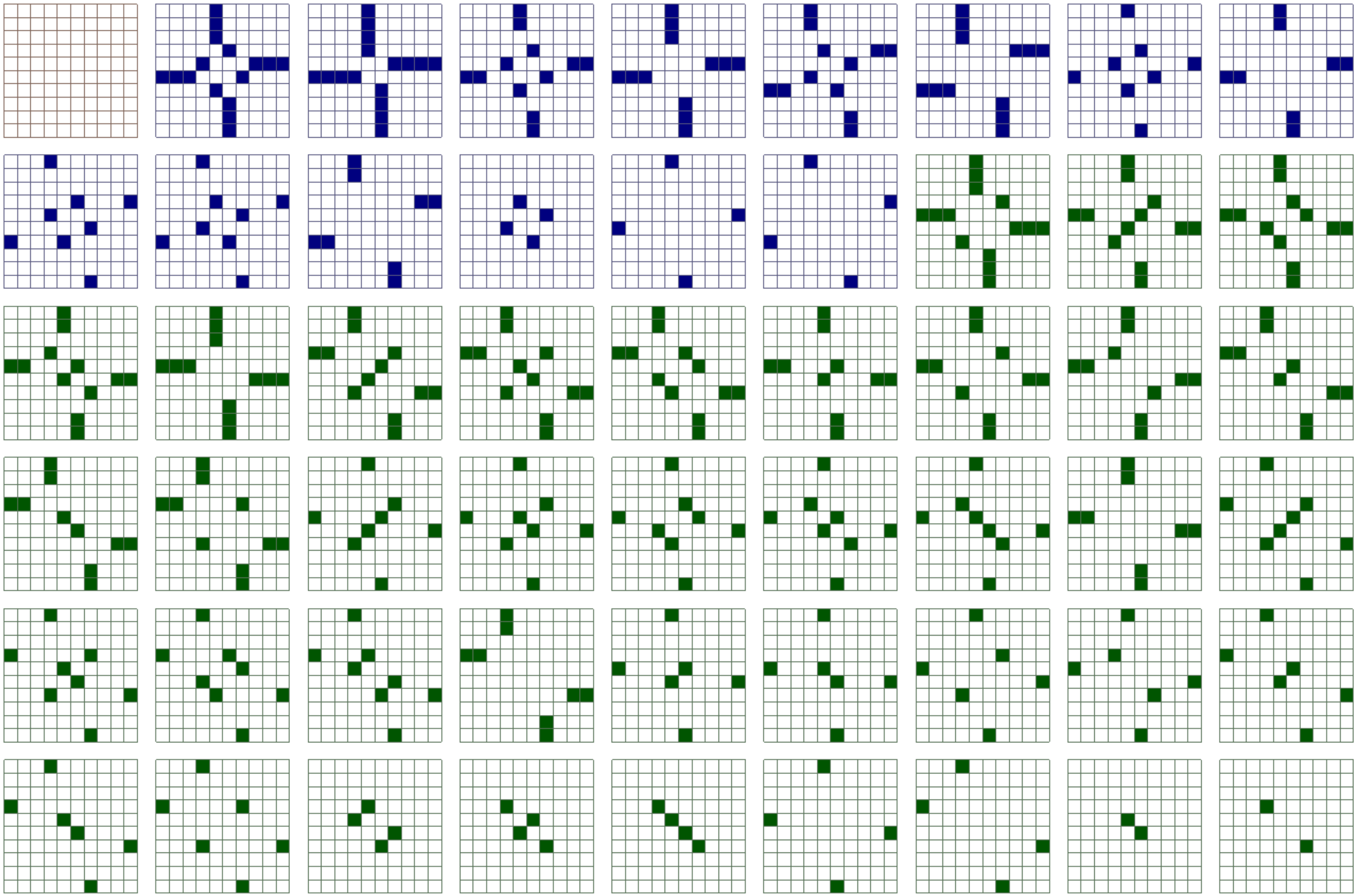
Maximal 9 x 9 Grids with w = 3



Maximal 10 x 10 Grids with w = 3



Maximal 10 x 10 Grids, w = 3, Black Omitted



Dynamic Programming Solution

- The Riddler asks for number of valid 15 x 15 grids with w = 3
- Answer:
  - 404,139,015,237,875 (tight case)
  - 409,764,131,469,788 (loose case)

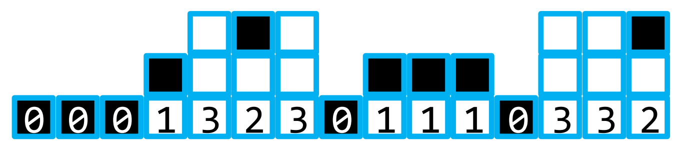
Warning: these take days to run

```
count[15,3,1] = 404139015237875
count[15,3,0] = 409764131469788
```

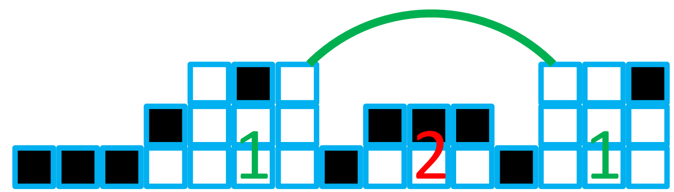
- Code to produce explicit lists of grids
  - Good for generating pictures on previous slides
  - Does not scale up to generating 400+ trillion grids!
- For the 15 x 15 case we use *dynamic programming*
  - Dynamic programming involves combining "overlapping subproblems"
  - For crossword counting, we define certain "states" and count how many partial grids (first j rows of full 15 x 15 solutions) are represented by each state
  - Counts of j x 15 solutions per state are updated to (j+1) x 15 counts
  - Instead of listing trillions of grids, update counts on millions of states

States

- Example: compute number of 15 x 15 tight grids with w = 3
  - After row 5 there are
    - 5,219,674 states encompassing
    - 6,984,251,647 partial solutions (5 x 15 grids)
  - One state is s = ((0,0,0,1,3,2,3,0,1,1,1,0,3,3,2), (1,2,1), True)
    - There are 3360 row 5 partial solutions for this state
- Meaning of state
  - (0,0,0,1,3,2,3,0,1,1,1,0,3,3,2) =



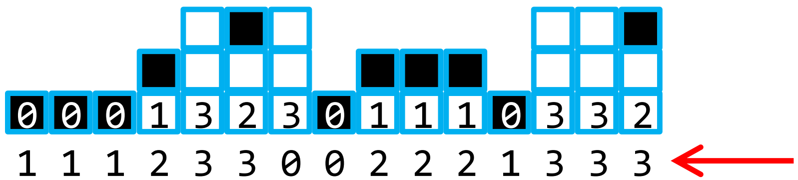
    - Heights of white-square stacks
    - But “3” means “3 or more”
  - (1,2,1) labels clusters of white squares

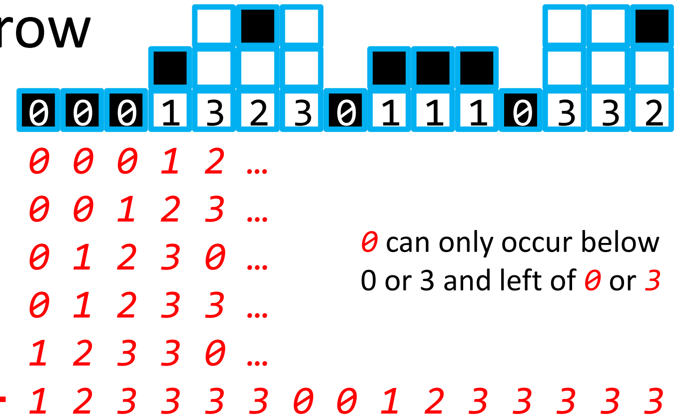


    - The state s represents partial grids with the first and last cluster connected
  - True: state s represents partial grids that have touched a side wall
    - This is needed to ensure that enumerated grids are tight

State Updates

- Generate all possible values of next row
  - Use temporary *horizontal numbering*
  - Convert each to vertical numbering





- Update cluster labels: (1,2,1) → (1,1)
  - Length 2 because 2 runs of positive numbers in new row
  - Same label (1) because both runs touch same cluster (1) in row above
- Update whether side wall has been touched (True → True)
- New state: ((1,1,1,2,3,3,0,0,2,2,2,1,3,3,3), (1,1), True)
  - Add 3360 to the row 6 count for this state
  - Add 3360 to the counts of all other states generated as above
    - From other red-numbered rows, provided they don't cut off a cluster

Final Step

- For each row 7 state s (representing many 7 x 15 partials G)
  - Think of forming C = G/R/G’ for any partial grid G represented by s
    - I.e., G atop a symmetric row R (with word lengths ≥ 3) atop G rotated 180°
  - Check that all vertical words in C have length ≥ 3
    - We can discern this from R and the vertical numbering in s
  - Check that C is connected
    - We can do this using cluster label information
  - Check that C is tight
    - If the last entry in s is “False”, require R to begin/end with white square
- Add up, over all row 7 states s,
  - The number of rows R that satisfy the above test, times
  - The count for state s
- This yields the total 404,139,015,237,875
  - Loose count can be obtained as sum over various tight cases

Number of Tight Valid n x n Grids for Various w

- Results of `count[n,w,1]`
  - Number of tight valid n x n grids with all word lengths ≥ w
  - Many values stored in CrosswordCounts package
    - Access precomputed values using `tightSquareCount[n,w]`
    - Cases that are not precomputed return the value −1

w \ n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	11	47	1411	21 411	2 123 851	124 511 195	42 999 348 087	9 581 445 219 371	11 829 924 769 055 787	−1	−1	−1	−1	−1
2	0	1	3	10	59	484	7250	181 575	6 826 137	446 562 953	43 131 669 850	7 112 223 095 914	−1	−1	−1	−1
3	0	0	1	3	12	48	312	2190	31 187	586 731	17 438 702	654 057 540	40 575 832 476	3 115 321 983 734	404 139 015 237 875	−1
4	0	0	0	1	3	12	50	208	1336	9119	113 415	1 993 875	37 724 992	1 290 193 576	45 949 047 420	−1
5	0	0	0	0	1	3	12	50	210	880	5971	38 421	487 427	7 583 483	126 501 143	−1
6	0	0	0	0	0	1	3	12	50	210	882	3694	25 455	165 436	2 079 601	−1
7	0	0	0	0	0	0	1	3	12	50	210	882	3696	15 442	109 248	−1
8	0	0	0	0	0	0	0	1	3	12	50	210	882	3696	15 444	64 348

`Table[tightSquareCount[n,w],{w,8},{n,16}]//MatrixForm`

- CrosswordCounts contains routines to check that efficient `count[m,n,w,t]` function agrees with simple, reliable `nValidGrids[m,n,w,t]`

Number of Tight Valid m x n Grids with w = 3

- Results of `countMat[15,3,1]`
  - Number of tight valid m x n grids with w = 3 for all 1 ≤ m,n ≤ 15
  - Stored in CrosswordCounts package
    - Access precomputed values using `countMat[3,1]`
    - Only w = 3 case as been precomputed

m \ n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
4	0	0	1	3	5	7	7	13	15	33	37	75	89	175	211
5	0	0	1	5	12	22	34	64	115	259	468	980	1742	3606	6519
6	0	0	1	7	22	48	86	178	367	973	2132	5456	11 520	28 508	60 985
7	0	0	1	7	34	86	312	626	1754	3990	12 931	30 741	100 126	234 362	749 439
8	0	0	1	13	64	178	626	2190	6746	21 972	66 411	234 143	738 990	2 613 144	8 061 187
9	0	0	1	15	115	367	1754	6746	31 187	108 089	472 654	1 599 558	7 463 594	25 710 394	119 869 867
10	0	0	1	33	259	973	3990	21 972	108 089	586 731	2 542 446	13 038 460	57 394 116	298 289 312	1 325 038 289
11	0	0	1	37	468	2132	12 931	66 411	472 654	2 542 446	17 438 702	85 912 738	571 540 158	2 794 999 844	18 851 755 888
12	0	0	1	75	980	5456	30 741	234 143	1 599 558	13 038 460	85 912 738	654 057 540	4 163 555 192	30 763 310 300	196 071 884 168
13	0	0	1	89	1742	11 520	100 126	738 990	7 463 594	57 394 116	571 540 158	4 163 555 192	40 575 832 476	287 957 992 192	2 772 709 489 316
14	0	0	1	175	3606	28 508	234 362	2 613 144	25 710 394	298 289 312	2 794 999 844	30 763 310 300	287 957 992 192	3 115 321 983 734	28 813 678 808 484
15	0	0	1	211	6519	60 985	749 439	8 061 187	119 869 867	1 325 038 289	18 851 755 888	196 071 884 168	2 772 709 489 316	28 813 678 808 484	404 139 015 237 875

`countMat15[3,1]//MatrixForm`

Number of Loose Valid m x n Grids with w = 3

- Results of `countMat[15,3,0]`
  - Number of loose valid m x n grids with w = 3 for all 1 ≤ m,n ≤ 15
  - Stored in CrosswordCounts package
    - Access precomputed values using `countMat[3,0]`
    - Only w = 3 case as been precomputed

m \ n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8
4	1	1	2	4	7	11	14	24	29	57	66	132	155	307	366
5	1	1	3	7	16	30	51	95	167	355	636	1336	2379	4943	8899
6	1	1	3	11	30	66	123	257	505	1263	2674	6794	14 283	35 477	75 479
7	1	1	4	14	51	123	398	814	2268	5064	15 668	36 786	117 537	274 755	873 496
8	1	1	4	24	95	257	814	2638	7942	25 616	76 522	265 290	827 121	2 907 117	8 949 504
9	1	1	5	29	167	505	2268	7942	35 325	120 281	521 379	1 751 561	8 086 842	27 699 924	128 712 668
10	1	1	5	57	355	1263	5064	25 616	120 281	635 325	2 731 307	13 913 459	60 876 022	314 844 598	1 394 036 694
11	1	1	6	66	636	2674	15 668	76 522	521 379	2 731 307	18 446 135	90 275 325	597 551 756	2 911 223 532	19 569 933 470
12	1	1	6	132	1336	6794	36 786	265 290	1 751 561	13 913 459	90 275 325	681 249 133	4 311 975 232	31 745 490 572	201 717 020 072
13	1	1	7	155	2379	14 283	117 537	827 121	8 086 842	60 876 022	597 551 756	4 311 975 232	41 752 489 853	295 090 915 631	2 833 434 360 883
14	1	1	7	307	4943	35 477	274 755	2 907 117	27 699 924	314 844 598	2 911 223 532	31 745 490 572	295 090 915 631	3 178 131 715 745	29 306 174 768 955
15	1	1	8	366	8899	75 479	873 496	8 949 504	128 712 668	1 394 036 694	19 569 933 470	201 717 020 072	2 833 434 360 883	29 306 174 768 955	409 764 131 469 788

`countMat15[3,0]//MatrixForm`

Summary

- CrosswordCounts has simple routines for generating
  - All valid crossword grids of minimum word length w
  - The subset of these valid grids that are maximal
- It includes routines for plotting sets of grids
  - Typically show only one grid from each “grid class” for brevity
  - Grids are colored by symmetry type
- A more sophisticated dynamic programming approach
  - Provides counts for cases too large to list solutions explicitly
  - Only counts number of valid grids
    - Not the number of grid classes
    - Not the number of maximal grids
  - Has been verified in two ways
    - Self-consistency: m rows of width n agrees with n rows of width m
    - Consistency with simpler, explicit approach