Further Insight into the Mondrian Art Problem

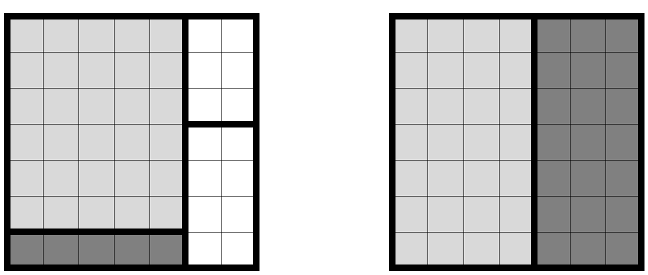
Hannes Bassen
18069 Rostock, Germany - hannesbassen@yahoo.de
(Dated: 29.11.2016)

In this work a computational method is presented, which allows to obtain the least possible defect in the Mondrian Art Problem. The found solutions are also optimal regarding the least used rectangles. We show optimal solutions up to a grid size of $n = 32$.

I. INTRODUCTION

Mondrian Art refers to the Dutch painter Piet Mondrian, who is known for the common use of rectangular shapes in his works. A deduced math problem, the Mondrian Art Problem, is introduced in [3] and is shortly presented here.

The situation is the following: We consider an equally spaced two-dimensional grid of size[1] $n \times n$. The whole area of the square has to be filled with non-overlapping non-congruent rectangles with integer dimension, the trivial case of one square filling the entire area is not allowed. Each filling of the square can be rated by the difference between the area of the largest and the smallest rectangle, which is denoted as *defect d* of the solution. The problem is to find the solution of grid size n with the smallest possible defect.



(a) (b)

Fig. 1: Exemplary solutions for the grid size $n = 7$. The defect of solution (a) is $d = 25$, for (b) it is $d = 7$.

Examples for solutions of the Mondrian Art problem are shown in fig. 1 for the grid size $n = 7$: The solution in fig. 1a consists of four rectangles with sizes 6×5, 4×2, 3×2 and 5×1. Obviously the defect of this solution is $6\times 5 - 5\times 1 = 25$, which is far from being optimal. An easier and better solution is shown in fig. 1b consisting of the rectangles 7×4 and 7×3, which defect is therefore $d = 7$.

As a consequence the solution in fig. 1b can be generalized to squares with odd grid size $n = 2m+1$. Partitioning the square into the two rectangles $n \times (m+1)$ and $n \times m$ leads to the least defect being $d \leq n$. However since squares with even size $n = 2m$ cannot be partitioned the same way, we divide them into $n \times (m-1)$ and $n \times (m+1)$ instead and receive the estimate $d \leq 2n$. As will be shown below there is still room for improvement. In the next section a computational method is presented, which allows to obtain optimal solutions for moderate grid sizes.

[1] Only square grids of different sizes are considered, therefore the grid size is simplified to n as of now.

II. THE ALGORITHM

A naive ansatz, one could come up with to find the optimal solution for a given grid size, is to fill the square with every possible rectangle at every possible position. Due to the immense amount of combinations this is not feasible. Often rectangles would not fit in the remaining free space either due to their area being too large or simply by being blocked by other rectangles. In order to face this issue, we present a 4-step algorithm below.

Step 1: Getting rectangles

First of all a list is created containing all non-congruential rectangles, that fit inside the square of size n (again, the trivial case of a $n \times n$ rectangle is not allowed). However all rectangles with an area larger than about half of the square area can be omitted - they would only increase the defect to $d > n$. This list is then sorted in descending order by the area of the rectangles.

Step 2: Getting recipes

Taking the list of all possible rectangles, we can generate a second list, each entry containing a list of rectangles, whose areas added together yield the square's area n^2. In this work these entries are refered to as *recipes*. In order to reduce the amount of memory and computing time needed to build this list, one can limit the allowed defect d to $d_- \leq d \leq d_+$. In practice setting d_- to zero and d_+ slightly above the expected defect is sufficient. If d_+ is set too low, the found recipes might not contain a valid solution.

Step 3: Removing impossible recipes

Now we have a fairly large list of recipes, each entry restricted to the square's area n^2. But lots of the recipes are impossible due to the fact, that their dimensions do not add up to n, meaning that the corresponding rectangles cannot fit inside the square without overlapping. By focusing on each rectangle in each recipe and checking in both directions, if there is a subset of the remaining rectangles, whose side lengths added to the width/height of the concerned rectangle yield the grid size, the impossible recipes can be removed from the list.

Step 4: Finding solutions

A list of recipes remains, which are restricted to both area and length. They are to this point valid candidates for solutions. A useful property of the list of recipes is, that its entries are sorted in increasing order primarily by the defect and secondarily by the number of rectangles necessary for a solution. This is a result of the steps 1 and 2.

In order to find a solution to a recipe or to prove its unsolvability, we need to exhaustively fill in rectangles in the square. Let's take a look at a situation, where a few rectangles are already filled in. At first we list all free concave corners and pick one. Then we try to insert one of the remaining rectangles in any direction inside the picked corner. As a major optimization it is not necessary to check the other found corners. In case one of the rectangles has no intersection with other rectangles, we insert it and again list all free concave corners etc. Otherwise, if no rectangle fits inside the free space in this step, the last inserted rectangle needs to be either turned in its corner or replaced by the next remaining rectangle.

Once all rectangles are successfully inserted, our solution is found, otherwise the last rectangle in the very first corner of the square needs to be replaced, but since all other rectangles were already checked, no solution exists. In the worst case for k rectangles there are roughly $2^k k!$ permutations, which need to be checked. However due to the applied restrictions to the recipes the average case is much better.

n	d	k	composition
3	2	3	2×2 3×1 2×1
4	4	4	3×2 4×1 2×2 2×1
5	4	3	5×2 3×3 3×2
6	5	5	5×2 3×3 6×1 3×2 5×1
7	5	5	6×2 4×3 5×2 4×2 7×1
8	6	6	7×2 6×2 4×3 5×2 8×1 4×2
9	6	3	6×5 9×3 6×4
10	8	6	10×2 5×4 6×3 8×2 7×2 6×2
11	6	8	9×2 6×3 8×2 4×4 5×3 7×2 6×2 4×3
12	7	10	9×2 6×3 8×2 4×4 5×3 7×2 12×1 6×2 4×3 11×1
13	8	6	8×4 10×3 6×5 9×3 13×2 8×3
14	6	6	6×6 7×5 11×3 8×4 10×3 6×5
15	8	5	12×4 8×6 15×3 11×4 8×5
16	8	8	6×6 7×5 11×3 16×2 8×4 10×3 6×5 14×2
17	8	12	14×2 7×4 9×3 13×2 5×5 12×2 8×3 6×4 11×2 7×3 10×2 5×4
18	8	10	18×2 12×3 9×4 7×5 11×3 16×2 15×2 6×5 14×2 7×4
19	8	10	10×4 8×5 13×3 19×2 12×3 9×4 7×5 11×3 16×2 8×4
20	9	10	9×5 11×4 14×3 7×6 20×2 10×4 13×3 12×3 9×4 6×6
21	9	10	16×3 12×4 8×6 15×3 9×5 11×4 14×3 7×6 8×5 13×3
22	9	12	15×3 9×5 22×2 11×4 14×3 7×6 10×4 19×2 18×2 12×3 9×4 6×6
23	8	7	16×5 10×8 11×7 19×4 18×4 12×6 9×8
24	9	11	19×3 14×4 8×7 11×5 18×3 13×4 17×3 10×5 7×7 16×3 12×4
25	10	14	25×2 10×5 7×7 16×3 8×6 23×2 22×2 11×4 21×2 14×3 7×6 20×2 10×4 8×5
26	9	13	19×3 14×4 8×7 11×5 18×3 26×2 13×4 17×3 10×5 7×7 24×2 16×3 12×4
27	10	16	25×2 10×5 7×7 24×2 16×3 12×4 8×6 23×2 15×3 9×5 22×2 11×4 14×3 7×6 20×2 10×4
28	9	15	19×3 28×2 14×4 8×7 11×5 18×3 9×6 26×2 17×3 25×2 10×5 7×7 24×2 16×3 12×4
29	9	11	9×9 16×5 10×8 26×3 13×6 11×7 19×4 25×3 18×4 12×6 9×8
30	11	15	11×6 16×4 8×8 21×3 9×7 30×2 20×3 15×4 12×5 10×6 19×3 28×2 14×4 8×7 11×5
31	≥ 11		
32	10	10	18×6 12×9 15×7 26×4 17×6 25×4 20×5 10×10 11×9 14×7

Tab. I: Number of rectangles k, defect d and composition of the solution for grid sizes up to $n = 32$.

III. RESULTS

With the described computational method at hand, we unleashed it on the grid sizes $n = 3$ to $n = 32$ and found several unknown solutions with low defects. Keeping the design of the algorithm in mind, the solutions are expected to be optimal with regards to the smallest possible defect and secondarily with regards to a composition with the fewest rectangles. All found best solutions are shown in tab. I: As before, n is the grid size, d the defect and k the number of rectangles used in the solution. Additionally the composition of each solution is displayed, the visual proof to these solutions can be found in the appendix. The found least possible defects are always less than or equal to the results, that were formerly known[1][2][3]. For $n = 31$ no solution with defect $d = 11$ is found yet.

By way of example the lowest known defect for $n = 18$ was reduced from 10[2] to 8 in this work, for $n = 14$ from 9[2] to 6. Although there seems to be a slight increase

d	n_1		n_2
0	3	-	419
1	3	-	700
2	4	-	897
3	3	-	417

Tab. II: Ranges $n_1 \leq n \leq n_2$ of grid sizes for small defects d, where no solution exists.

in the defect for larger grid sizes, no obvious pattern is visible.

Another result, this algorithm gives by adjusting the parameters d_- and d_+, is that there are no solutions with defect $0 \leq d \leq 3$ in certain ranges of grid sizes, as is shown in tab. II. Solutions for these defects (with the exception of $d = 2$ for $n = 3$) might be rare, nonexistent or only found at larger grid sizes.

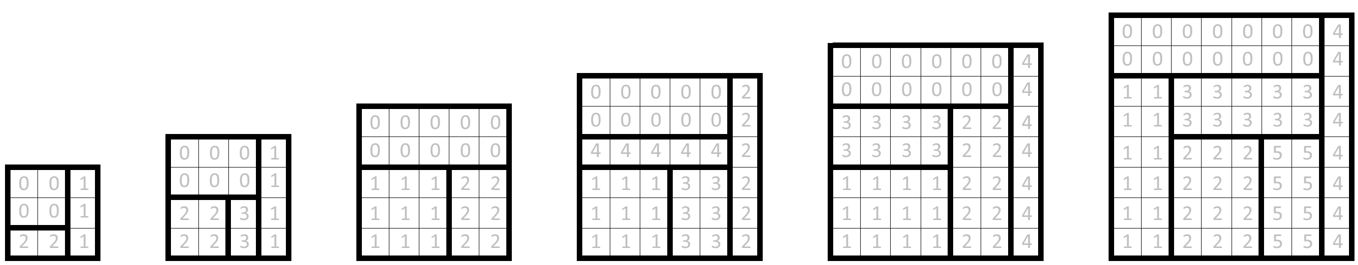
IV. CONCLUSION

In this work we presented a performant algorithm to solve the Mondrian Art Problem, which is capable of sorting out impossible solutions and quickly searching for a solution given a valid recipe. Now the least possible defects are known for grid sizes less than 31 and for $n = 32$. Even though a lot of time went into tweaking the algorithm, it runs into trouble regarding computing time, if the number of rectangles in a recipe is larger than about 16 to 17. This happens quiet frequently for larger grids beyond the ones shown in tab. I. Nonetheless the new algorithm can find (all) valid candidates for a solution with least possible defect, although the actual search for a solution might not be computable at this point in time with the current method, its implementation and the used hardware.

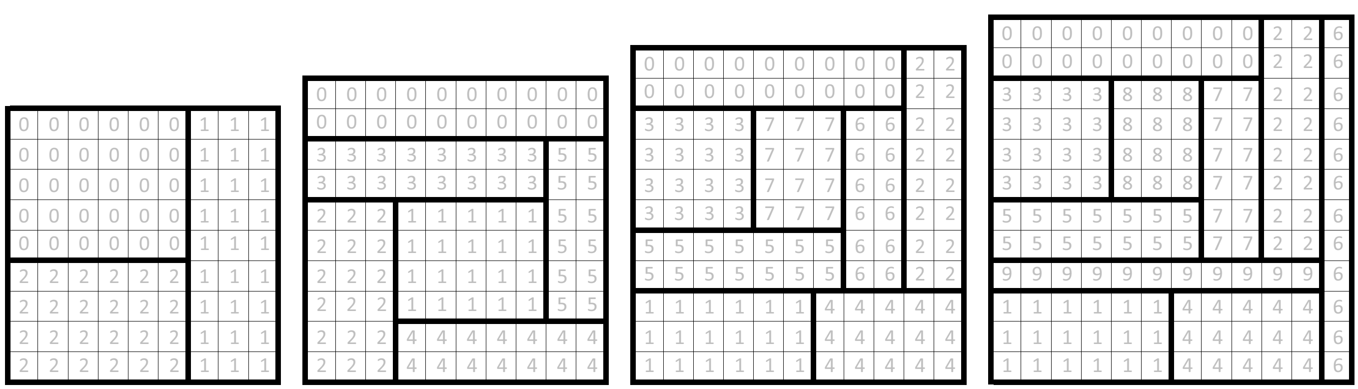
[1] E. Pegg Jr, *Mondrian Art Problem*, `http://demonstrations.wolfram.com/MondrianArtProblem` (visited on 25.11.2016)
[2] E. Pegg Jr, sequence A276523, in: OEIS, `https://oeis.org/A276523` (visited on 25.11.2016)
[3] G. Hamilton, *Mondrian Art Puzzles*, `http://mathpickle.com/project/mondrian-art-puzzles` (visited on 25.11.2016)

APPENDIX

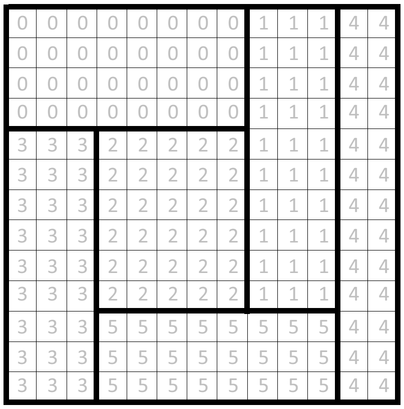
Below all solved squares corresponding to the solutions listed in table I are shown. Smaller integers inside the squares denote a larger area of the rectangle and vice versa - zero denotes the biggest rectangle.



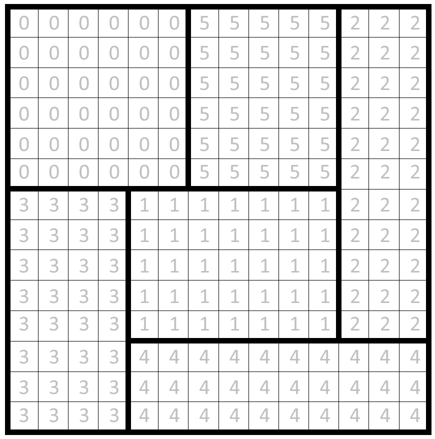
n=3 n=4 n=5 n=6 n=7 n=8



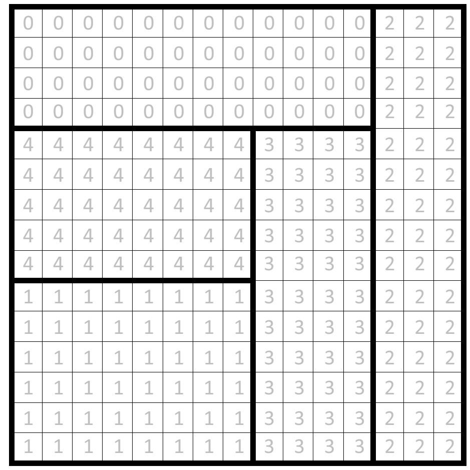
n=9 n=10 n=11 n=12



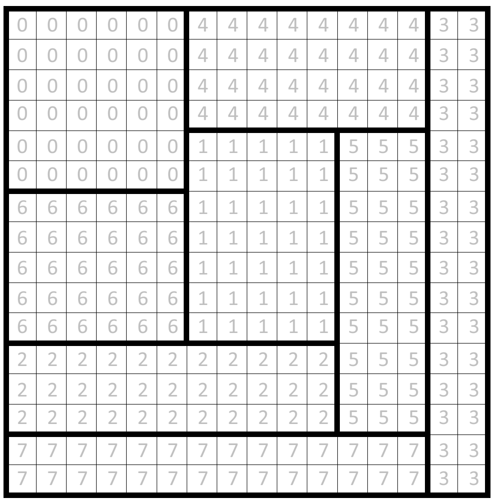
n=13



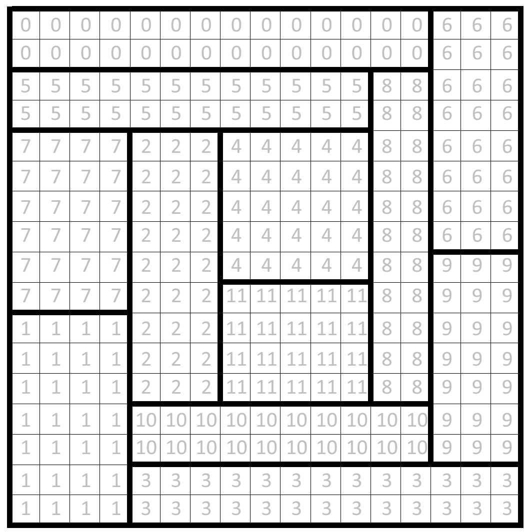
n=14



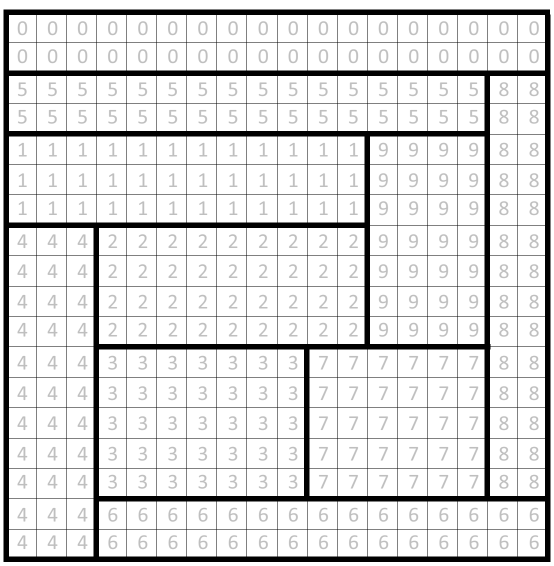
n=15



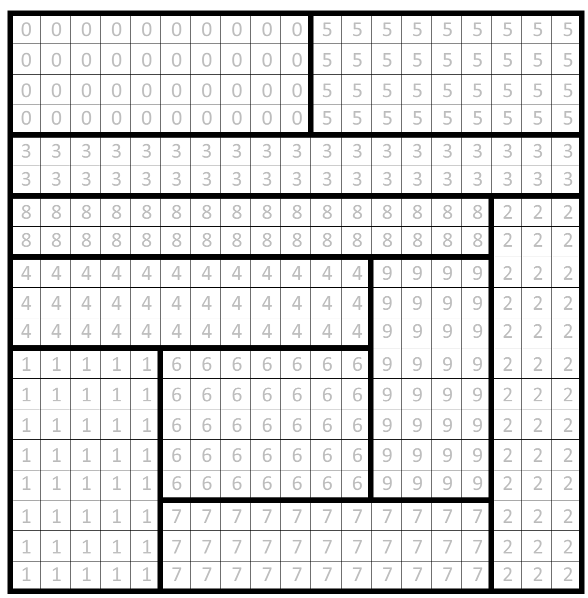
n=16



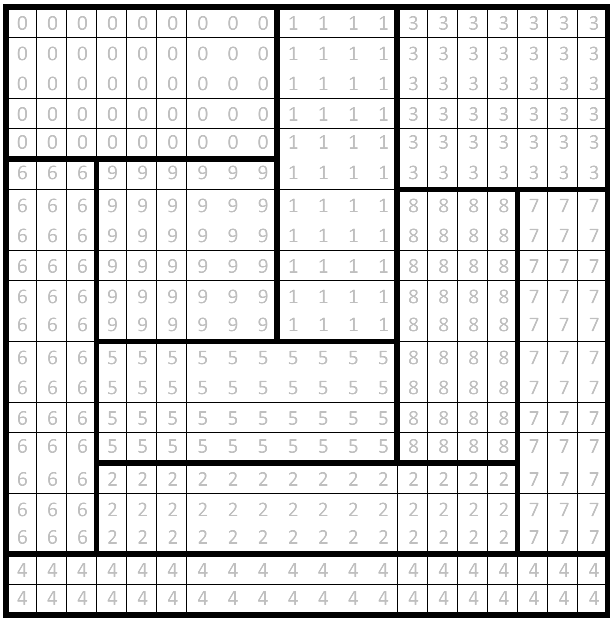
n=17



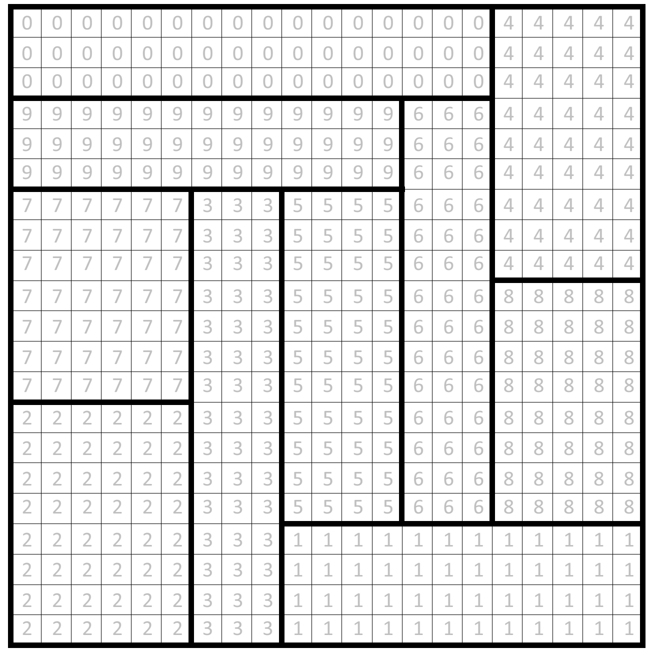
n=18



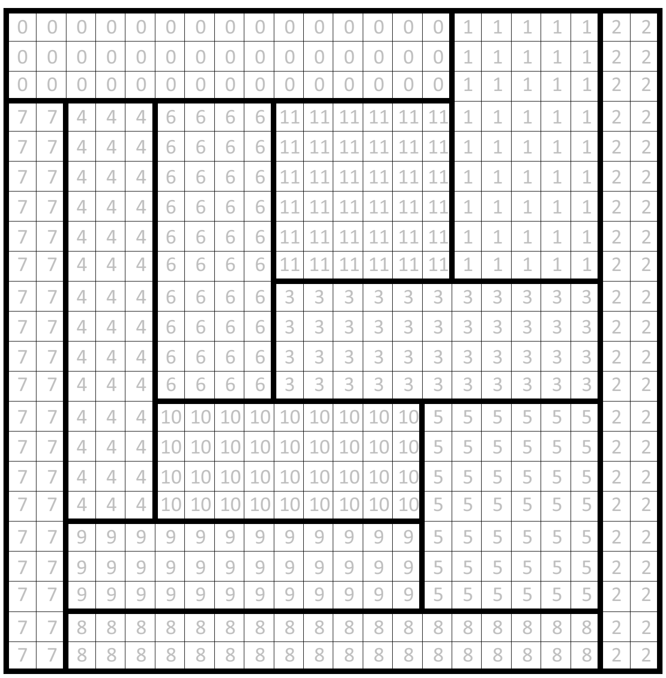
n=19



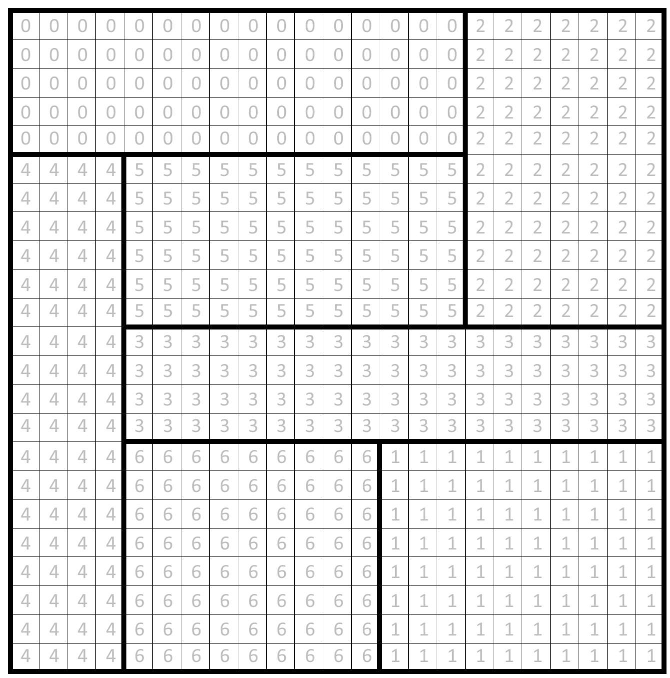
n=20



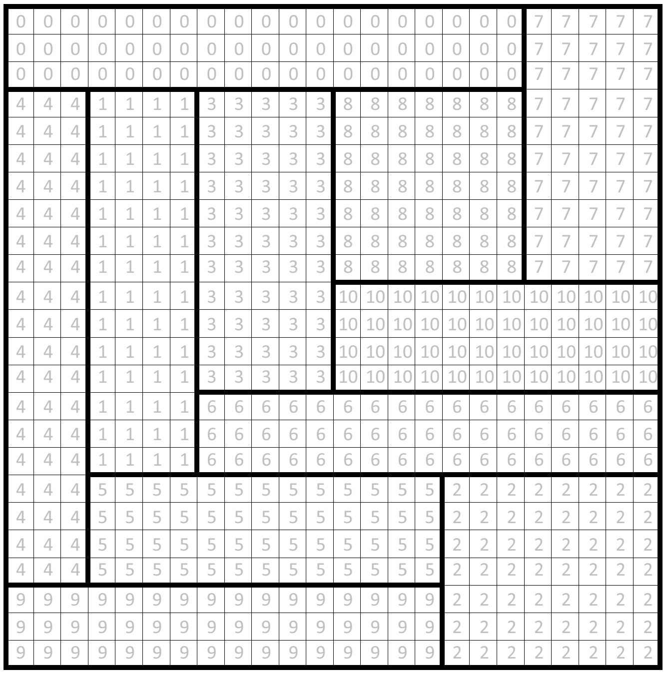
n=21



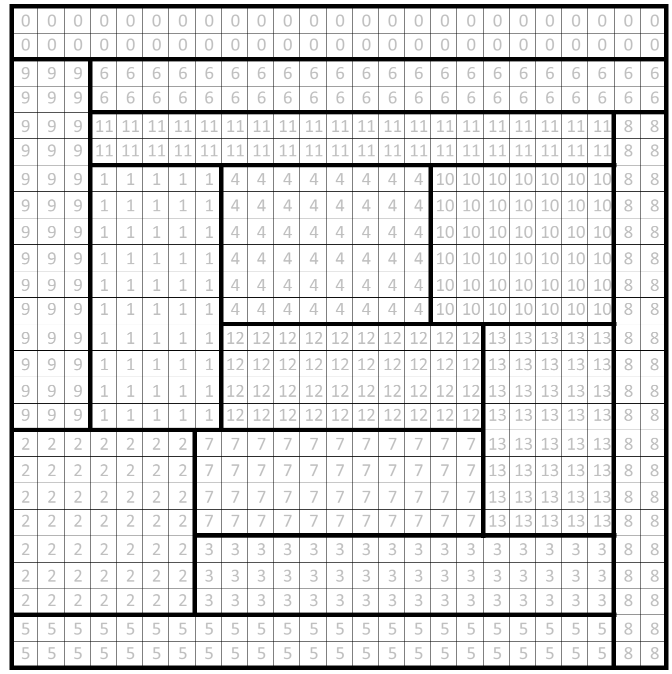
n=22



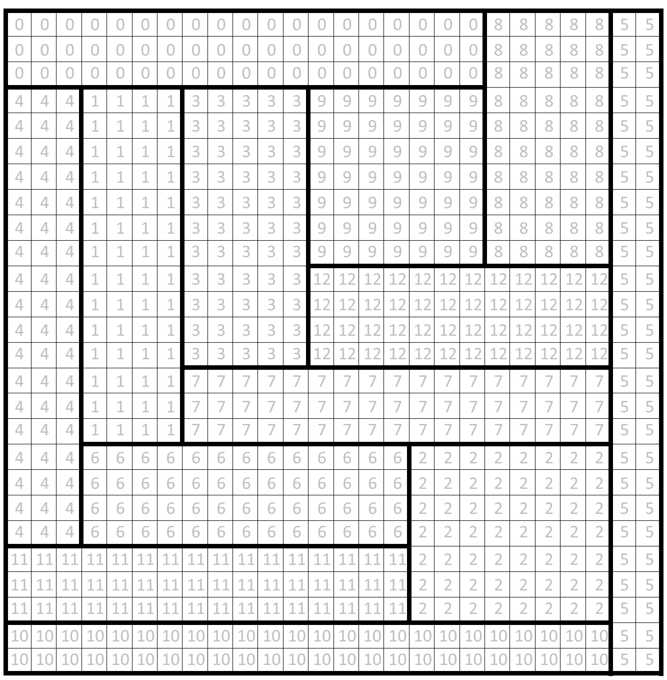
n=23



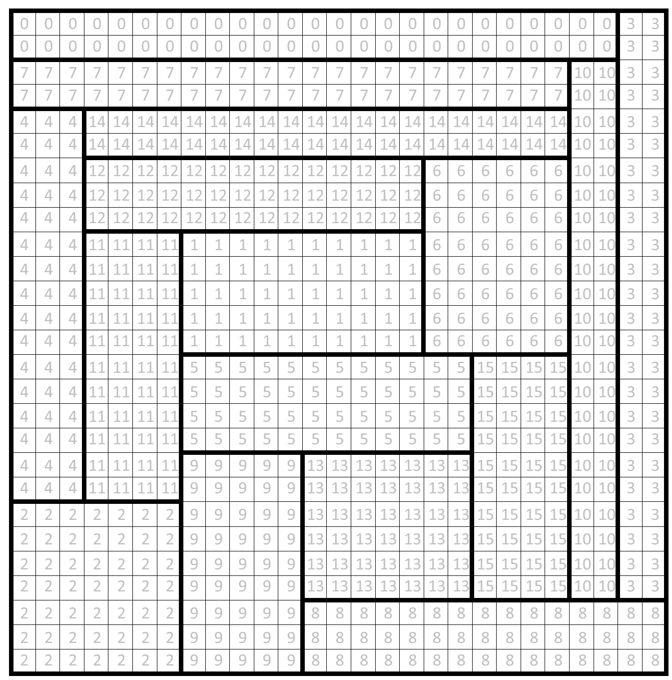
n=24



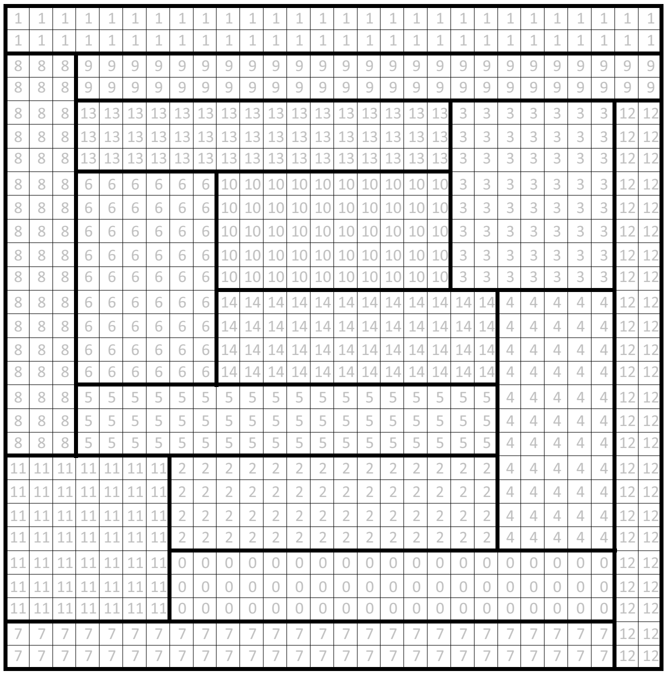
n=25



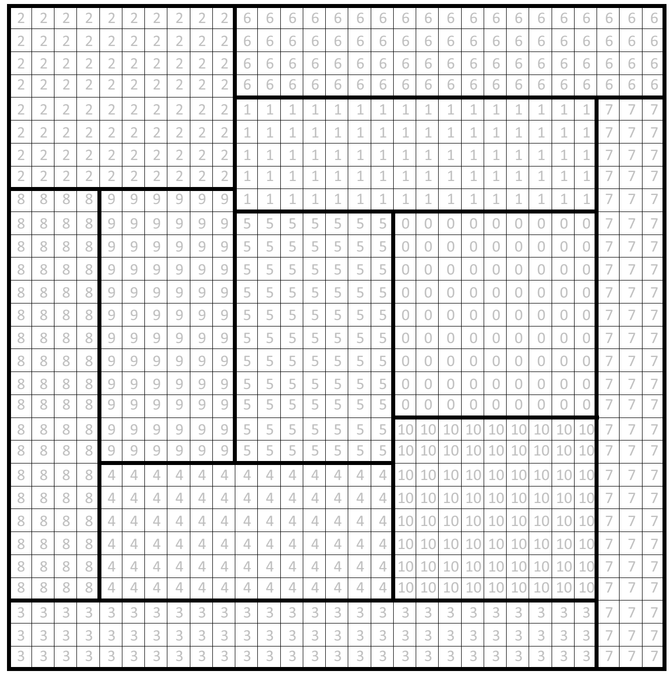
n=26



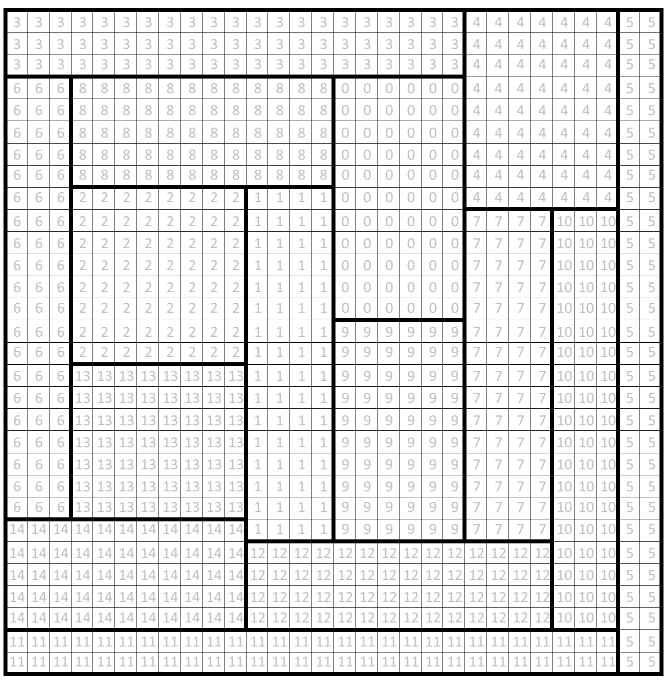
n=27



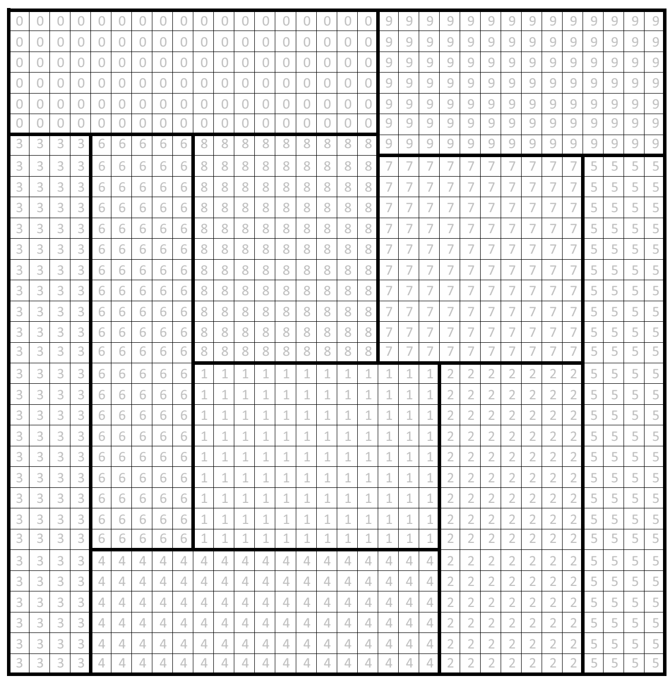
n=28



n=29



n=30



n=32