

Chapter Exercises

5.1 / 5.2

See code in attached triominos.py file.

5.1 and 5.2 use the same code and tile the solution according to the given algorithm. The location of the deficient square is randomized when the board is first initialized. Below are the outputs of boards with a size of 2, 4, and 8.

The numbers represent the pieces and their respective tiling iteration. For example, a triomino with the number “1” was tiled first.

Size 2^1	1 total pieces (1, 1) UR <pre> 1 1 X 1 </pre> Time to Tile Board: 1.5974044799804688e-05 Process Complete
Size 2^2	5 total pieces (2, 2) LR (1, 3) LL (1, 1) LL (3, 3) UR (3, 1) LR <pre> 2 X 4 4 2 2 1 4 3 1 1 5 3 3 5 5 </pre> Time to Tile Board: 6.413459777832031e-05 Process Complete
Size 2^3	21 total pieces (4, 4) LL (2, 6) UL (1, 7) UL (1, 5) LL (3, 7) UR (3, 5) UL (2, 2) LR (1, 3) UL (1, 1) LL (3, 3) LL (3, 1) LR (6, 6) UL (5, 7) UL (5, 5) LL (7, 7) UR (7, 5) UL (6, 2) LR (5, 3) LR

```

(5, 1) LL
(7, 3) UR
(7, 1) LR

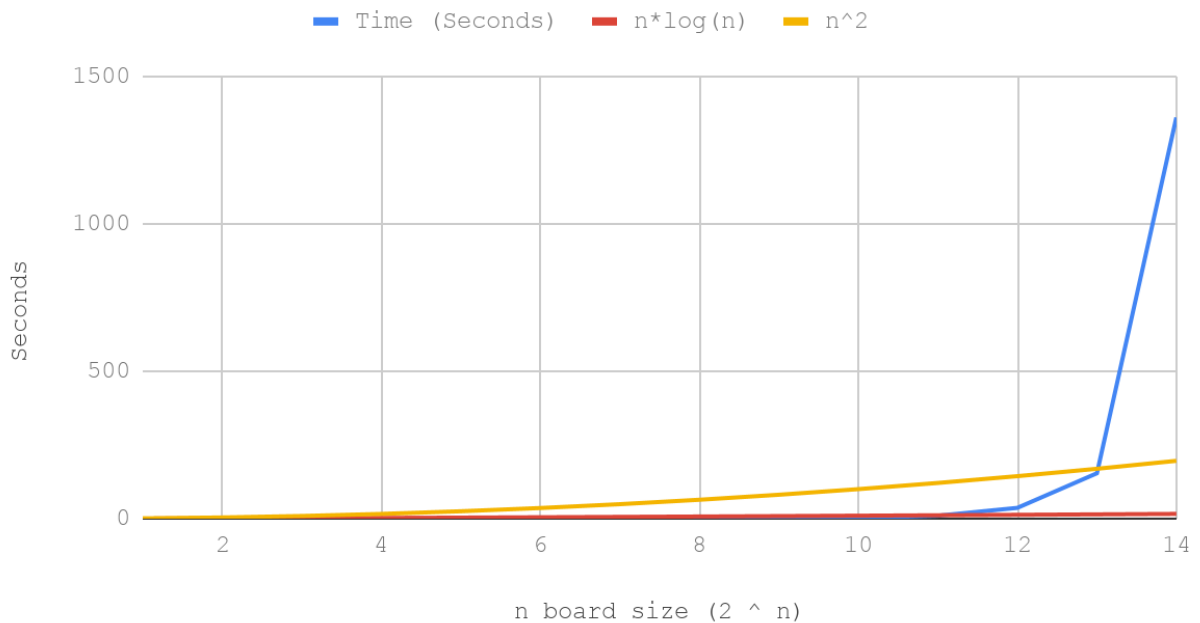
03 03 05 05 13 13 15 15
03 02 02 05 13 12 12 15
04 02 06 06 14 12 16 16
04 04 06 01 14 14 16 16
08 08 10 01 01 18 20 20
08 08 07 10 18 18 17 20
09 07 07 11 19 17 17 21
09 09 11 11 19 19 21 21

Time to Tile Board: 0.00022792816162109375
Process Complete

```

The process also prints out the board in a graphical manner, fulfilling the 5.2 requirements. The algorithm can tile any board as long as it is a power of 2. There is a check on this requirement at line 22. The final output printed in the process states the time it took to tile this particular board. Below is an analysis of the time complexity compared to $O(n \log n)$ and $O(n^2)$.

Tiling Time vs. $n \log(n)$ vs. n^2



The data for the chart is given here. The algorithm is more efficient than $n \cdot \log(n)$ for the first 11 iterations, or up to a board size of 2048 x 2048. After this, the algorithm is O inefficient compared to $n \cdot \log(n)$ but more efficient than n^2 . However, after the 13th iteration (8192 x 8192), the algorithm becomes more inefficient than n^2 . The maximum tested board has been 2^{14} (16384 x 16384) and this test run takes about 22 minutes.

n	Time (Seconds)	$n \cdot \log(n)$	n^2
1	1.79E-05	0	1
2	4.29E-05	0.6020599913	4
3	0.0001559257507	1.431363764	9
4	0.0006868839264	2.408239965	16
5	0.002575874329	3.494850022	25
6	0.00937795639	4.668907502	36
7	0.02914619446	5.91568628	49
8	0.1141731739	7.224719896	64
9	0.454417944	8.588182585	81
10	2.008261919	10	100
11	9.966018677	11.45531954	121
12	36.49931693	12.95017495	144
13	153.7590511	14.48126358	169
14	1360.05978	16.0457925	196