Algorithmics	Student information	Date	Number of session
	UO: 284185	1/3/2022	3
	Surname: Fernández-Catuxo Ortiz		
	Name: Rita		

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. Tromino Times.

N	T TROMINO
4096	94
8192	341
16384	1295

These are the values I obtained from a reliable time until a heap overflow.

1. **What should be the time complexity of the algorithm?**

As the problem is solved by division (we divide the board in 4 quadrants), we must apply one of these formulas:

- $O(n^k)$ if $a < b^k$
- $O(n^k * \log n)$ if $a = b^k$
- $O(n^{\log_b a})$ if $a > b^k$

a = 4 as we do four recursive calls. The number of sub problems (number of quadrants we divide the board) is 4.

b = 2 because each time we make a recursive call we divide by 2 the size of the board.

k = 0 because excluding recursive calls, the complexity of the method is O (1).

Then, as 4 > 2^0 (a > b^k), we have a complexity of O (n ^log2 4) = O (n^2).

Algorithmics	Student information	Date	Number of session
	UO: 284185	1/3/2022	3
	Surname: Fernández-Catuxo Ortiz		
	Name: Rita		

2. **Check if the time obtained in the previous section does or does not meet the theoretical complexity of the algorithm.**

According to the following results, we can conclude that the values obtained meet the theoretical complexity of the algorithm:

We know that **t2 = (f(n2) / f(n1)) x t1**, being $f(n) = n^2$

Taking this values:

n1 = 4096 t1 = 94

n2 = 8192 t2 = 341

$t2 = (8192^2/4096^2) \times 94 = 376 \approx 341$

Taking this values:

n1 = 8192 t1 = 341

n2 = 16384 t2 = 1295

$t2 = (16384^2/4096^2) \times 341 = 1364 \approx 1295$