


Algorithmics	Student information	Date	Number of session
	UO: 277921	16/3/2021	4
	Surname: García López	 Escuela de Ingeniería Informática Universidad de Oviedo	
	Name: Rosa		

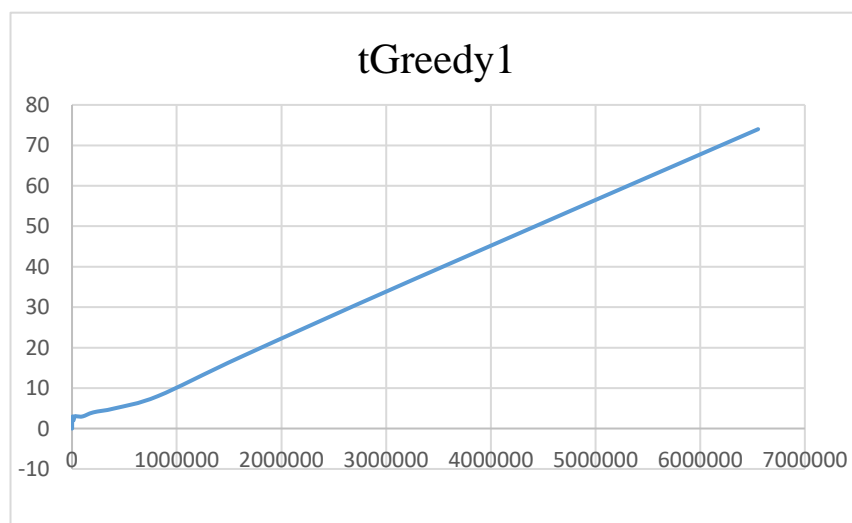


Universidad de Oviedo  
Universidá d'Oviéu  
University of Oviedo

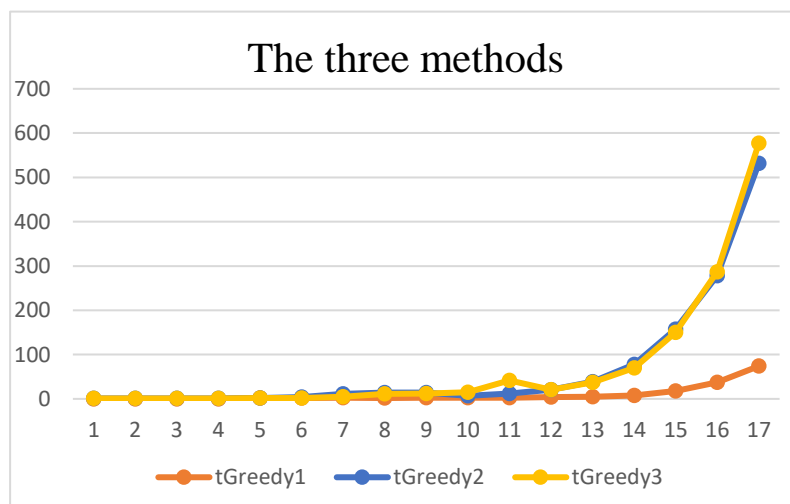
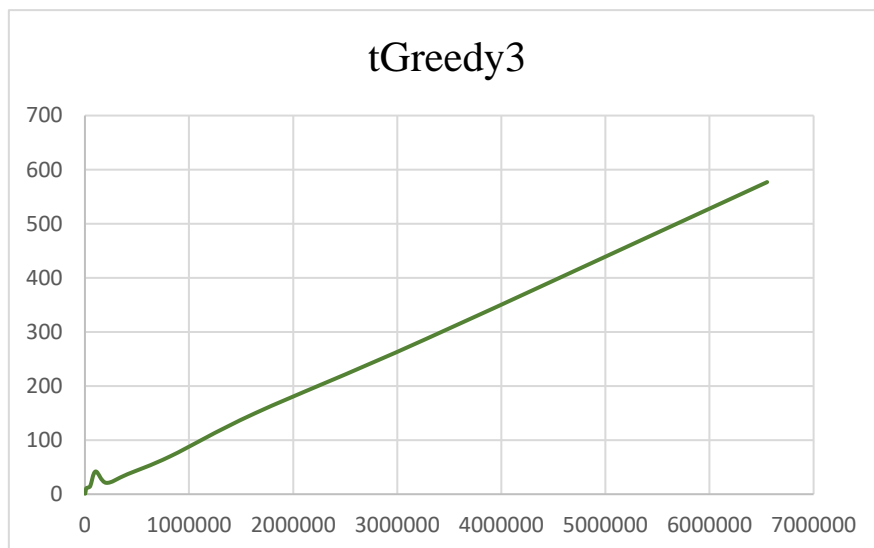
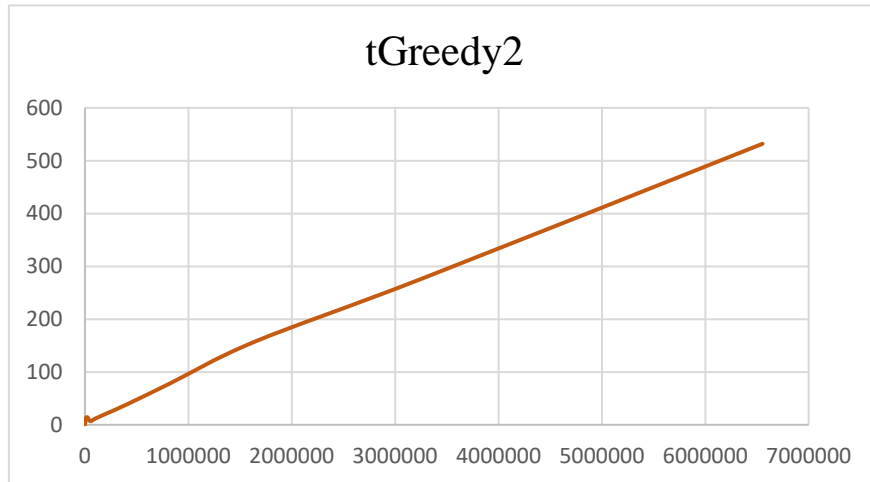
## Activity 1. Execution times

The time is in milliseconds.

$n$	$t_{Greedy1}$	$t_{Greedy2}$	$t_{Greedy3}$
100	0	1	1
200	0	1	1
400	0	1	1
800	0	1	1
1600	2	2	2
3200	2	4	2
6400	3	11	5
12800	2	14	11
25600	3	14	12
51200	3	7	15
102400	3	12	42
204800	4	21	21
409600	5	39	37
819200	8	78	70
1638400	18	157	150
3276800	37	278	287
6553600	74	532	577



Algorithmics	Student information	Date	Number of session
	UO: 277921	16/3/2021	4
	Surname: García López		
	Name: Rosa		



Algorithmics	Student information	Date	Number of session
	UO: 277921	16/3/2021	4
	Surname: García López		
	Name: Rosa		

## Activity 2. Answer the following questions

- 1. Explain if any of the greedy algorithms involves the optimal solution from the point of view of the company, which is interested in maximizing the number of “pufosos”.**

Yes, the second solution maximizes the result of “pufosos”. The results obtained on each solution for the file *game1.txt* are: Greedy1 = 428.5, Greedy2 = 477.5, and Greedy3 = 168.5. Thus, from the point of view of the company the best algorithm is the Greedy2.

- 2. Explain if any of the greedy algorithms involves the optimal solution from the point of view of the player, who is interested in minimizing the number of “pufosos”.**

Yes, the third solution minimizes the result of “pufosos”. As we have seen from the results before, from the point of view of the player the best algorithm is the Greedy2.

- 3. Explain the theoretical time complexities of the three greedy algorithms, according to the implementation made by each student, depending on the size of the problem  $n$ .**

In my implementation the *Greedy1* algorithm has a time complexity  $O(n)$ , and, due to calls to methods of external libraries of Java, I do not know the exact complexity of *Greedy2* and *Greedy3* algorithms; but they are both similar and more complex than  $O(n)$ .

- 4. Explain if the times obtained in the table are in tune or not, with the complexities set out in the previous section.**

From the table where the practical data have been collected, we can calculate a expected time  $t_2$  for a  $O(n)$  complexity. Then:

$$t_1 = 8 \text{ ms}, n_1 = 819200 \text{ and } n_2 = 1638400; t_2 = \frac{n_2}{n_1} * t_1, t_2 = 16 \text{ ms}$$

Algorithmics	Student information	Date	Number of session
	UO: 277921	16/3/2021	4
	Surname: García López		
	Name: Rosa		

The result obtained theoretically is a bit less than the one practically obtained, nevertheless it is really approximated so we can conclude that the complexity is  $O(n)$  and we are right.

As I am not able to know the complexity of the remaining algorithm, I cannot calculate anything theoretically; we can only make use of the results obtained and compare them through them. Therefore, I can only say certainly that the *tGreedy3* seems to be more complex than *tGreedy2* but both of them have a similar complexity.