Activity 1. Execution times

The time is in milliseconds.

|  |  |  |  |
| --- | --- | --- | --- |
| ***n*** | ***tGreedy1*** | ***tGreedy2*** | ***tGreedy3*** |
| 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 |

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.

1. **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.

1. **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 , 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 .

1. **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 *t2* for a complexity. Then:

t1 = 8 ms, n1 = 819200 and n2 = 1638400; , t2 = 16 ms

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 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.