ALGORITHMICS - Bachelor of Software Engineering, 2021

LAB GUIDE. SESSION 4

GOALS:

- **Greedy algorithms**

1. Segment placement

The company **JuegosSencillosSL** has just launched a simple game where there are n segments (s_0, s_1, s_2, …, s_{n-1}), each one with its corresponding length (l_0, l_1, l_2, …, l_{n-1}). Lengths are integer numbers in the interval [1..99].

The player must choose as the first segment the one she wants and put it on a line starting from the 0 coordinate. Then, she must choose as the second segment the one she wants (of the n-1 remaining segments) and place it after the first, and so on until she uses/takes all the segments.

The placement of each segment is defined by three variables: where the segment begins (x), where the segment ends (y), and the midpoint (x+y)/2.

In the end and, once the n segments have been placed, the player must pay a virtual amount (in a cryptocurrency called “PUFOSA”), the well-known pufosos, which will be equal to the sum of all the average positions (midpoints) of the n placed segments.

It goes without saying that the player is interested in ensuring that the amount to be paid is as low as possible and that the company is interested in ensuring that the amount to be paid as high as possible.

Well, to see how to play and place the segments, we consulted three top experts in ALGORITHMICS and each of the three gave us the following advice (curiously, all three are greedy heuristics):

GREEDY 1: It consists in the ostrich algorithm, that is, "go placing the segments in the same order in which they are in the file, because in the end every solution gives the same cost, then why are we going to complicate our lives?".

GREEDY 2: If consists in placing them from longest to shortest length.

GREEDY 3: If consists in placing them from shortest to longest length.

2. Example

Let us see an example. For example, a file game1.txt with the following content:

```
6
20
60
4
8
4
17
```

The content of the file means that we have 6 segments (first line) with the following lengths: 20, 60, 4, 8, 4, 17

Geedy 1:

It will calculate:

S0: (0 to 20), midpoint=10

S1: (20 to 80), midpoint =50

S2: (80 to 84), midpoint =82

S3: (84 to 92), midpoint =88

S4: (92 to 96), midpoint =94

S5: (96 to 113), midpoint =104.5

Cost of greedy 1 = 428.5 pufosos

Geedy 2:

It will calculate:

S1: (0 to 60), midpoint =30

S0: (60 to 80), midpoint =70

S5: (80 to 97), midpoint =88.5

S3: (97 to 105), midpoint =101

S2: (105 to 109), midpoint =107

S4: (109 to 113), midpoint =111

Cost of greedy 2 = 507.5 pufosos

<u>Geedy 3</u>:

It will calculate:

S2: (0 to 4), midpoint =2

S4: (4 to 8), midpoint =6

S3: (8 to 16), midpoint =12

S5: (16 to 33), midpoint =24.5

S0: (33 to 53), midpoint =43

S1: (53 to 113), midpoint =83

Cost of greedy 3 = 170.5 pufosos

TO DO:

A. Work to be done

- An `algstudent.s4` **package** in your course project. The content of the package should be:
  - A class `SegmentsPlacement.java` that will expect as an argument a text file with the game. The text files should be in the same package.
    - It will basically have a method to read the game file (input of the game) and three methods (one for each one of the greedy algorithms). It is requested to **print the solution proposed by each greedy algorithm together with its associated cost**. Logically, apart from that, you can implement all the methods you need. The **main** method should include all the needed code to execute the 3 greedy algorithms, one after the other. The main method should also have an argument: the path of the text file to be used.
  - A class `SegmentsPlacementTimes.java` that will not expect any argument.
    - This class will simulate (through the `Random` class) inputs of different sizes `n` and for each `n` they will measure the time it takes to execute each of the 3 greedy algorithms.
    - Obviously, if you notice that it is necessary to use `nTimes`, you should use it.
    - You should also exclude from the times you take the time you need to print the solutions (`System.out.println`).
- A **PDF document** using the course template. The activities of the document should be the following:
  - **Activity 1. Execution times.**

- The goal is to complete the following table of times:

n	*tGreedy1*	*tGreedy2*	*tGreedy3*
100	……….	……….	……….
200	……….	……….	……….
400	……….	……….	……….
800	……….	……….	……….
1600	……….	……….	……….
3200	……….	……….	……….
6400	……….	……….	……….
12800	……….	……….	……….
…	……….	……….	……….

- **Activity 2. Answer the following questions.**
  - 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".
  - 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".
  - 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`.
  - Explain if the times obtained in the table are in tune or not, with the complexities set out in the previous section.

You should include in your Java project a new `algstudent.s4` package with the following content inside it:

- All the requested source files.
- The requested PDF document called `session4.pdf` with the corresponding activities.

Deadlines:

- The deadline is one day before the next lab session of your group.