Homework 3
CSE 246 Analysis of Algorithms, Spring 2017
Due 19.05.2017 Friday 23:59 (no extension - strict!!!)

In this homework you are asked to design and code efficient algorithms for the following two problems:

Problem 1: Consider the following modified version of the coin-collecting robot problem. Each cell has an integer value between -5 and 5. If the value of a cell is positive, then the robot gains that amount of coins when it visits that cell. If the value of the cell is negative, then the robot pays that amount of coins if it visits that cell and if it has enough coin. But if it has not collected that amount of coins in the previous cells, then the robot cannot visit a cell with negative value. Also there are some cells on the board that are always inaccessible for the robot.
Again the robot starts at (1,1), it can only move towards right and down and it stops at (*n*,*m*). Following corresponds to an example board, where inaccessible cells are shown by X's and values of cells are shown by integers between -5 and 5:

0	-1	5	4	0	0
1	2	0	X	4	0
3	1	3	-1	X	5
0	0	-2	3	-1	5
X	X	X	0	1	1

Now you are asked to do the following:

- **a)** Given an *n* x *m* board with all the values and accessiblity information, describe an algorithm that finds the following:
  i. Maximum number of coins the robot bring to cell (*n*,*m*).
  ii. A path from (1,1) to (*n*,*m*) that results in maximum coin collection. If there are multiple such paths, your algorithm may return any of them.

  You need to have a pdf document file, **report1.pdf**. In this document, give a step by step verbal description of your algorithm in detail.
- **b)** Apply your algorithm for the above 5 x 6 board.
- **c)** Give time and space complexity of your algorithm.
- **d)** Implement your algorithm in Java. Input file name should be given as command line argument.

  Input file format should be as follows. The first line will contain two integers *n* and *m*. What follows will be the *n* lines, each line corresponds to each row of board (that includes integer values between -5 and 5, and X's). All the values will be seperated with tabs (\t).

  Output file format should be as follows. First line should include an integer value for the number of coins collected by the robot. What follows will be the n lines, each line corresponds to each row of the output board, where the path is shown with P. As an example, for the following input:

```
5    6
0    1    -2   5    X    4
2    -2   2    4    X    0
3    1    0    -1   1    1
3    -1   -2   X    2    5
X    X    -5   0    1    1
```

Contents of the output file will be the following:

```
14
P    1    -2   5    X    4
P    P    P    P    X    0
3    1    0    P    P    1
3    -1   -2   X    P    P
X    X    -5   0    1    P
```

Your grade will depend on:

1. How clear you describe your algorithm;
2. Whether your algorithm works correctly on our inputs or not. (So format is very important).

Problem 2: Suppose that there are n laptops each containing a wireless transmitter. For each laptop i, following information are known:

- Position, (x_i, y_i),
- Wireless transmission range, r_i

That is, we can imagine that the wireless range of laptop i is a circle centered at (x_i, y_i) with radius r_i. We say that the laptop i can communicate with laptop j if laptop j is in the wireless range of laptop i. (If wireless range of laptop j is less than wireless range of laptop i, it is possible that laptop i can send message to laptop j, but laptop j may not be able to send message to laptop i.) Of course, not every laptop can communicate with every other laptop, but laptops can send messages by using intermediate laptops as routers. Hop distance $h(i, j)$ is defined as the minimum number of intermediate laptops used to send a message from laptop i to laptop j. For example, if two laptops can communicate directly, the hop distance between them is 1. Now, you are asked to do the following:

a) Given a set of n agents with their positions and wireless ranges, design an efficient algorithm to compute the hop-distance from the first laptop to every other reachable laptop. If agent i is not reachable from agent j then the hop distance $h(i, j)$ will be set to 0. You need to have a pdf document file, **report2.pdf**. In this document, give a step by step verbal description of your algorithm in detail.

b) Give time and space complexity of your algorithm.

c) Implement your algorithm in Java. Input file name should be given as command line argument.

Input file format should be as follows. The first line will contain an integer n, which is the number of agents. What follows will be the n lines, where the i'th line contains three real numbers x_i, y_i and r_i corresponding to the x and y coordinate of the i'th laptop, along with its wireless range. All the values will be separated with tabs (\t).

Output file format should be as follows. There will be n lines. On the i'th line will be an integer, representing the hop distance $h(1,i)$, the minimum number of intermediate laptops necessary for the first agent to communicate with agent i.

Your grade will depend on:

1. How clear you describe your algorithm;
2. How efficient your algorithm is (and whether you indicate efficiency correctly);
3. Whether your algorithm works perfectly on our inputs or not. (So format is very very important).

Note 1: Please submit your commented source codes, some sample input files and output files (that are generated by your code), **report1.pdf** and **report2.pdf** in a zip file that includes both your name(s) and surname by email to cse246submit@gmail.com. In the zip file, files related to problem 1 and problem 2 should be in different folders.

Note 2: Plagiarism will be severly punished. Note that, there are very powerful tools that finds similarity between two different codes. **We will use such tools.** So, **do not share your code with anyone else!**

Note 3: You may do this homework in groups of two. But not more!