CS 333 - Homework #2

Due To

1 May 2024 Wednesday, by 11.59 pm.

Homework 2

There are 2 independent tasks in this homework.

Task1 (50 Points)

Problem Description

The input to our algorithm is going to be n total boxes. Each box has a length, width, and height. We will always call the largest dimension the box's length and the smallest dimension the box's height (i.e. for each box, $length \geq width \geq height$).

For two boxes, A and B, box A fits inside of box B provided A's length is strictly less than B's length, A's width is strictly less than B's width, and A's height is strictly less than B's height. For example, a box with dimensions $(20, 19, 18)$ fits inside a box with dimensions$(20.1, 19.5, 19)$, but it does not fit into a box with dimensions $(20, 20, 20)$.

Given a list of n boxes, each with a length, width, and height, you are to write a dynamic programming algorithm which prints the maximum nesting depth for that collection of boxes. In other words, your algorithm should print just a single integer representing the the largest possible number of boxes that can be nested. The filename for source code will be "box.py" or "Box.java".

Input

A list of n box objects, each box object has a length, width, and heigh attribute.

Output

Your algorithm should print just the integer representing the maximum nesting depth. You algorithm should have no other print statements. Input will be given as a file by providing the filename as a command line argument. To provide an input with n items, you will have $n + 1$ lines in your input as follows:

- The first line contains the value of n, i.e. the number of boxes in the list.

- The remaining n lines contain a space-separated list of 3 floats/doubles, each box's dimensions. The first number (which is the largest of the three) represents the length, the second represents the width, the third (which is the smallest) represents the height.

Python Sample run: python box.py test1.txt

Test Cases Summary

There are 5 test cases provided to you. The answers for these tests are listed below:

- test1.txt has answer 5
- test2.txt has answer 9
- test3.txt has answer 8
- test4.txt has answer 15
- test5.txt has answer 16

Task2 (50 Points)

Problem Description

A ceiling fan is blowing dust around a square room in a regular pattern. At the same time, a robot vacuum is following a path through the room. Your task is to see if the robot will be able to capture the dust, and if so, how long it will take.

You are given a grid that divides the room into cells. Each cell has one of the following directional symbols: '>', '<', '^', and 'v'. The robot always starts in the cell at the upper, left corner. At each timestep, it moves to the adjacent cell indicated by the direction of the symbol in its current cell. The robot will never be directed to move outside of the room.

The dust moves in a periodic pattern through the room. You are given its position at each timestep in one iteration of the pattern. After it completes an iteration, it repeats the pattern. If the robot and the dust are in the same location at the same timestep, the robot vacuum will capture the dust.

Input

The input begins with an integer r on a line by itself, that gives the length of one wall of the room. Since the room is square, the width is equal to the length.

The next r lines give the pattern that the robot will follow in the room. Each line will have r symbols, chosen from the following: '>', '<', '^', and 'v'.

Next there is an integer d which gives the length of the pattern that the dust follows.

The following d lines contain two space-separated integers. Starting at $i = 0$, the line i contains (r_i, c_i) indicating that at timestep i the dust will be in row r_i and column c_i. Note that the first row is numbered 0, and the first column is numbered 0. Also, after the dust is in the last position in the input, the pattern repeats and it is moved back to (r_0, c_0).

Output

Output consists of single line. The line should read "never" if the dust will never be captured by the robot vacuum. Otherwise, it should contain the number of timesteps needed for the robot to capture the dust.

Python Sample run: python robot.py test1.txt

Constraints

- $1 \leq T \leq 10$
- $3 \leq r \leq 400$
- $1 \leq d \leq 160,000$
- $0 \leq r_i, c_i < r$ for $0 \leq i < d$

Test Cases Summary

There are 3 test cases provided to you. The answers for these tests are listed below:

- test1.txt has answer 6
- test2.txt has answer never
- test3.txt has answer 14

Submission Rules

You will submit this homework via the LMS system. You should follow the file-naming conventions and guidelines below:

- You should submit your source files as a ZIP archive file (NOT RAR or other formats). The name of the file should be in format **"<USER-ID>_hw<HOMEWORK-NR>.zip"**. For example, if your username is vy1043, then the name of the submitted file should be "vy1043_hw2.zip". Pay attention that all the letters are in lower-case. ZIP archive is supposed to contain just the source files, under two folders corresponding to the two tasks ("Task1"and "Task2" folders).
- Late submissions and files that do not compile are not accepted.
- You can resubmit your homework (until the deadline) if you need to.
- You are not allowed to use any external libraries and/or functions. Everything must be implemented by you from scratch.
- Any type of plagiarism will not be tolerated. Your submitted codes will be compared with other submissions and also the codes available on internet and violations will have a penalty of -100 points. (In case of copying from another student both parties will get -100)