Programming Assignment 5: Roomba

CECS 328

1 Deadline

December 10th, 2021 at 5 PM

2 Introduction

You are designing an algorithm for the Roomba Minus, an automated floor sweeper. Though the Roomba Minus is a major advance in robotic home maintenance, there are some serious design flaws. The engineers are looking to you to help them improve the product.

- The Roomba Minus is designed to be placed somewhere along the wall of one side of a rectangular room, and will travel to the opposite wall of the room. If you draw the room in such a way that one wall lies along the line $y = 0$ and the opposite wall lies along the line $y = h$ (for some constant h), then the speed of the Roomba Minus in the y direction will always be constant (and positive). We will refer to this speed as v.
- The engineers have designed the Roomba Minus so that it can move horizontally as well. Given its vertical speed v, it can move horizontally at any speed between $-v/r$ and v/r (for some constant r) *as long as it is still moving vertically while it does so.* The horizontal speed is not constant and can be changed at any time.

Given a rectangular room, the Roomba Minus will first identify problematic points in the room and analyze their severity. It will then place itself on the south ($y = 0$) side of the room and work its way methodically to the other side of the room ($y = h$), cleaning up the dirty areas of the floor that it reaches as it goes. Given the restrictions that the engineers have designed into it, it's clear that the machine will not be able to handle every single problematic point in the room. Your goal is to design an algorithm for the Roomba Minus that will (a) specify where on the $y = 0$ axis it should start and (b) which problematic points it should clean up on its way to the other side of the room. The algorithm should maximize the sum of the severity points of the problematic points in the room that it cleans.

3 Your code

You will be given r (the ratio of vertical to horizontal speed). Following this will be n lines, each containing an x and y coordinate, the coordinates of the mess, and a severity s for the mess.

Your program will list off the messes that your Roomba Minus cleans in chronological order, one per line. Assume that the array of messes begins counting at 0.

HINT: It is possible to formulate this problem as the problem of finding the longest path in an appropriately chosen directly acyclic graph. Imagine putting an edge from one mess to another if and only if it is possible to reach the second from the first.

The Java header in StudentSolover.java will be:

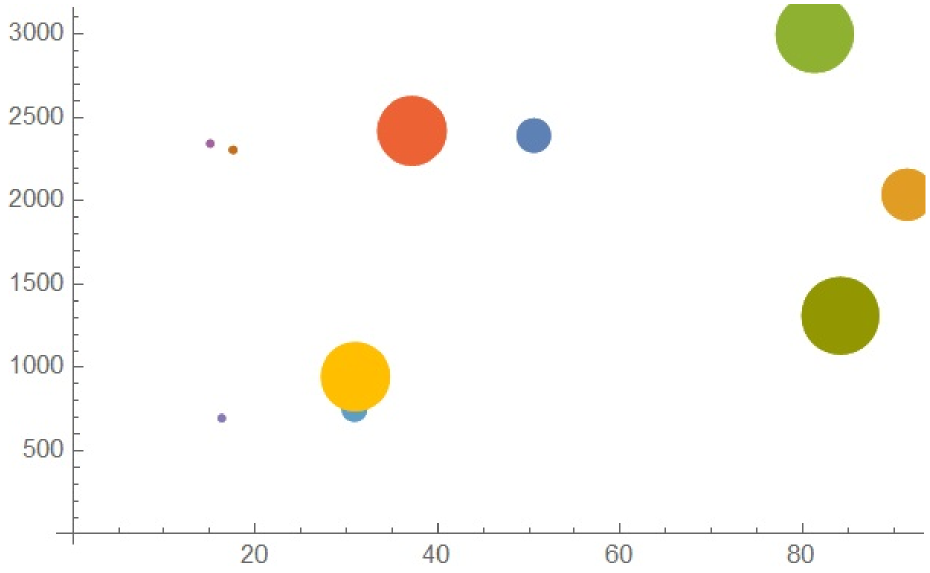
```
public static ArrayList<Integer> run(double r, ArrayList<Pair<Pair<Double,
Double>, Integer>> mess)
```

The Python header in studentsolver.py will be:

```
def run(r, mess)
```

The C++ header in StudentSolver.h will be:

```
static vector<int> run(double r, vector<pair<pair<double, double>,
int>>);
```



4 Example

Assume that you are given the following messes with ratio 16.7374586913063:
((50.53339753651755, 2392.9913262022305), 4),
((91.45350267880721, 2038.5690414517053), 6),
((81.30630354764358, 3002.1564113738523), 9),
((37.16228801441648, 2421.3261154038723), 8),
((16.323410763410276, 696.3036909610697), 1),
((17.558652660436632, 2306.3556734125145), 1),
((30.858752483887464, 751.748794481916), 3),
((30.957197204880696, 944.2879593578829), 8),
((15.042626905954554, 2344.979171359295), 1),
((84.13614550978521, 1310.743768220772), 9)
The optimal answer is

$$[6, 7, 1, 2]$$