CSCI 270 Fall 2019 Programming assignment 1

Instructor: Joseph Bebel

Due: Wednesday, October 23th, 2019, 4:59 pm

1 One and Two Day Delivery

A certain online store sells products with one and two day delivery in a city with streets laid out on a rectangular grid. The warehouse is located at address $(0, 0)$ and there is one delivery truck. The customers $1, \ldots, n$ are sorted in the chronological order that they paid. Customer i is located at coordinates (x_i, y_i). The store has promised that if customer i bought their item before customer j (that is, if $i < j$) *and* both customers i and j's purchases are delivered on the same day (i.e. both on day 1 or both on day 2), then the truck will always deliver to customer i before customer j.

The truck uses $|x_i - x_j| + |y_i - y_j|$ dollars of fuel to drive between points (x_i, y_i) and (x_j, y_j) on the grid. The store wants all of its customers to be satisfied, but delivering everyone's items on day 1 will use a lot of fuel. To offset this, the store decides to deliver only some purchases on day 1, and give a promotional store credit to customers whose purchases are delivered on day 2. Customer i requires c_i dollars to remain satisfied if their delivery happens on day 2.

The truck must start and end at $(0, 0)$ on both days.

Given as input the values (x_i, y_i, c_i) for each customer, output the minimum amount of fuel and credits that the store needs to spend to make all of the deliveries within 2 days and leave all customers satisfied.

1.1 I/O Format

1.1.1 Input

Your program should read from **standard input**. The first line contains a single integer n, followed by n lines describing n customers. The i-th line contains three integers x_i, y_i, c_i, separated by space.

1.1.2 Output

Output a single integer to **standard output** - the minimum amount of dollars to spend (fuel plus customer credits).

1.1.3 Sample I/O

Please see `resource/asnlib/publicdata/...` on *Vocareum*.

1.2 Data Range

- For 10% of full credit, pass all test cases with $n \leq 15$.
- For 75% of full credit, pass all test cases with $n \leq 100$. (Expected complexity: $O(n^3)$)
- For full credit, pass all test cases, $1 \leq n \leq 1000, -10^5 \leq x_i, y_i \leq 10^5, 0 \leq c_i \leq 10^5$. (Expected complexity: $O(n^2)$)

1.3 Implementation Details

You should submit a **single** source file containing the implementation of your solution via *Vocareum*, in one of the following programming languages:

C Filename should be `assignment1.c`. Compile flags: `-lm -std=c11`.

C++ Filename should be `assignment1.cpp`. Compile flags: `-std=c++11`.

Java Filename should be `assignment1.java` and the public class containing your `main` method should be named `Solution`. You should not declare any package.

Python Filename should be `assignment1.py`. The environment is `Python 3.6.4`.

The performance of your program is evaluated based on your actual run time. The time limit is **1 second** for C/C++/Java and **3 seconds** for Python.

1.4 Grading

We will grade based on the performance and correctness of your algorithm on a set of test cases. Your solution will run against sample test cases and a set of pretests upon submission where you can see the submission report, but they do **not** count towards your final score.

You will receive points of a test case if and only if your solution terminates within the time limit and your output is correct. If your algorithm is not correct (fails on a test case) or too slow, then you will get partial credit depending on what test cases your algorithm does pass.