Report for Homework Assignment 2

20175095 Jisu Shin

i) Elaborating problem statement and requirements in points

This problem consists of two big set of problems.
The first problem is to put the tarps in the order by given upper and bottom coordinates. On this phase, from the given requirement, we have to check out whether the given data are intersecting or invalid values.
After listing tarps on order, we can handle tarps just like a horizontal line.
The next one is to figure out the minimum number of punctures among lines to reach the vineyard which is the fundamental problem. So we should find the algorithm here to check along the lines and get the minimum number of puncutres.

ii) Data structure used and why it has been used

I used two vectors to get the input values. First one is to get the given inputs (x1, y1, x2, y2) and make them to two pairs ((x1, y1), (x2, y2)) so I declared vector<pair<pair<int, int>, pair<int, int>>> tarps. Second one is to get the tarps that are modified to the horizontal line after the sorting. So, it contains lower y, direction of the line, x1, x2 which is declared vector<pair<pair<int, int>, pair<int, int>>> tarps_new.
Mating them by the pair is necessary because as the lines are sorted by the y coordinates, coordinates of each lines(x1, x2) have to move together with it. Also vector is useful on this situation because we can handle each data more easily. So those data structures have been used on the code.

iii) Algorithm used to solve the problem and why

Brute-force algorithm is used on this problem. As this problem does not have time limit and data set size is small, we can solve this by brute-force algorithm.
First, vary the number of changing the directions of rain falls. The changing direction varies from 0 to the number of tarps so we use for loop here.
Inside the for loop, vary the reaching point of the rain to vineyard. The point varies from leftmost part of the vineyard to the rightmost part of the vineyard so we use another for loop here.
Inside this for loop, we will ascend from bottom to the upper tarp which is the reverse version of the rain falls down along the tarp. It is more effective to handle this problem by ascending back, not falling down because previous loop is about the coordinate of the vineyard which is easier to handle by ascending.

By those loops, all situations will be checked which is kind of brute-force algorithm and we can find out the minimum number of punctures to get the vineyard.
I tried a lot but I could not finish my code on this part so I left it as a blank for loop.

iv) Briefly explain an alternate solution, if there is

After sorting the tarps and change them to horizontal lines, we can use dynamic programming which is the original solution when time limit exists. Set the x-coordinates to whole array and change the value 0 when it is on the range of vineyard, otherwise infinite. As the rain goes down and gets the new line, the array value of the point increases 1 when we puncture the point there. This situation will reduce the running time because array keeps the value of the previous execution. When it reaches vineyard, the smallest number of the value becomes the minimum number of the punctures to get to the vineyard.

v) Calculation of time complexity of the algorithm used

As I wrote on third question, for loops are tripled on the main function and on the each loop, the index changes by 1. So time complexity will be about O(n^3).