Computer Science 320SC – (2016)

Assignment 3 (programming)

Due: Sunday, September 4th (9pm)

Requirements

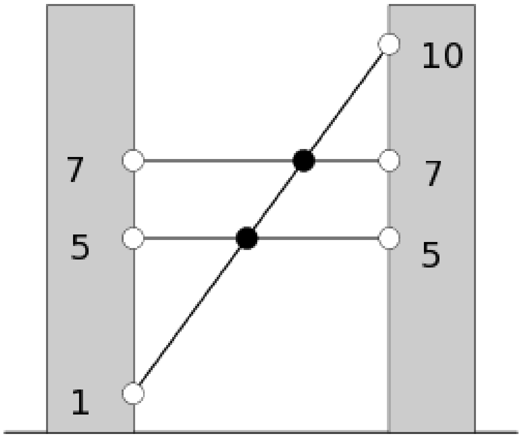
This assignment lets you get familar with greedy and divide-and-conqurer algorithm design. It is worth 5% of your total course marks.

Problem Statement: Intersecting Wires

This problem is taken from a recent `code.google.com/codejam` contest. Although not required at the time the problem specifications were given, we have modified a few of the constraints so you will (most likely) need to implement an efficient divide-and-conquer algorithm to pass the CS320 harder data set.

An intranet company wants to connect two buildings with many wires, each connecting a port on the first building to a port on the second building. Wires are straight segments connecting a vertical position on the left building to a vertical position on the right building. No two wires share the same endpoint on a building. However, from our side viewpoint, some of the wires intersect midway. We also noticed that due to different tensions of the wires exactly two wires meet at any given intersection point.

On the picture below, the intersection points are the black circles, while the ports are the white circles.



The question we want to know is how many intersection points should we see given, as input, a set of pairs of vertical distances that represent wire connection points on the two buildings.

Input

The first line of the input gives the number T of test cases. The T test cases follow. Each case begins with a line containing an integer N, denoting the number of wires you see.

The next N lines each describe one wire with two integers A_i and B_i. These describe the ports that this wire connects: A_i is the height of the port on the left building, and B_i is the height of the port on the right building.

Assume the following limits $1 \leq T \leq 1000$, $1 \leq A_i \leq 1000000$, $1 \leq B_i \leq 1000000$ and $1 \leq N \leq 50000$, Within each test case, all A_i and all B_i are different. No three wires intersect at the same point.

The input should be taken from keyboard/stdin/System.in. The output should be sent to console/stdout/System.out.

Sample Input:

```
2
3
1 10
5 5
7 7
2
1 1
2 2
```

Output

We have two problems. The first problem checks if you can read the input and compute a list of wires that are the longest in length. With x being the case number, output a single line containing "Case #x: " followed by a single white-space separated list in increasing order of wires (starting at index 1).

Sample Output (longest wires):

```
Case #1: 1
Case #2: 1 2
```

For the main problem, output one line containing "Case #x: y", where x is the case number and y is the number of intersection points you see from the side vantage point.

Sample Output (number of inversions):

```
Case #1: 2
Case #2: 0
```

Submission

For this assignment name your source code `wiresL.ext`, `wiresE.ext` or `wiresH.ext`, where `ext` denotes one of { `java`, `cpp`, `py` } that indicates java/c++/python language. Please use just one source file per problem. Here the suffix `L` of the basename denotes 'L'ongest wires, `E` dennotes 'E'asy (test data) and `H` denotes 'H' arder (test data). One mark is allocated to `wiresL.ext`, and two marks are allocated for both `wiresE.ext` and `wiresH.ext`.