

Practice 10 (2016/12/6)
Geometry problem starting from a LINE

1. Problem

Calculating the overlapping of line segments is the fundamental problem of computational geometry. In this problem, we consider the simplified line overlap problem – how to calculate the overlapping length of horizontal line segments.

In this problem, all horizontal lines have no y -coordinate, and they can be regarded to have the same y -coordinate. Thus the overlapping check can be achieved by only examining the x -coordinates of two endpoints of every line segment. You have to identify all two line pairs with overlap and sum up the total overlapping length. For example, there are three line segments and any two line segments overlap. Thus you have to find three overlapping lengths and then sum up these three overlapping lengths. Two-loop codes can simply solve this problem, but its time complexity is $O(n^2)$, where n is the number of line segments. The runtime will increase substantially as n increases. The time complexity of the optimal solution to this problem is $O(n \log_2 n)$. All benchmarks for this problem should be able to be solved within 0.156 second. Please design an algorithm to calculate the overlapping length of a horizontal line set and guarantee to solve each benchmark within 2 seconds (including 2 seconds). Any test for a benchmark over 2 seconds is regarded to fail.

Input Format:

There are two integers in a line, where the first integer is the x -coordinate of left endpoint and the second integer is the x -coordinate of right endpoint. Two integers are separated by space character. Notably, the number of space characters is not fixed, do not use space-sensitive instruction to read input file.

Input example 1: there are four horizontal line segments and their input content is as follows. $overlap_len(i,j)$ is the overlapping length of line i and line j . The total overlapping length for these four line segments is: $overlap_len(1,2) + overlap_len(1,3) + overlap_len(2,3) = 45 + 155 + 20 = 220$. Notably, the overlapping length between lines 1 and 4 is 0.

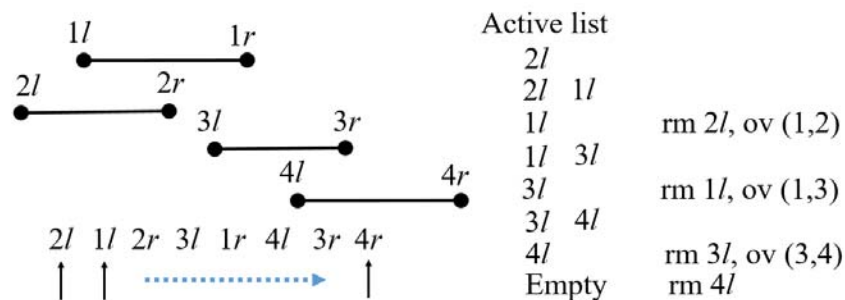
```
75  325
5   120
100 255
325 500
```

Output format: Please print out the total overlapping length of the line set.

Output example: the output content of the above example is as follows. Since the total overlapping length may exceed the maximum value of an integer, please use a

double-type number to output your result.

220



Problem 2 (100 pts)

- (a) Another method to compute the total overlapping length for a line set is to sort two endpoints of all lines in increasing order. In the sorted list, each point is attached with a attribute to indicate the left or right endpoint. Then we start to scan the sorted point list and perform the following operations:
1. If current point is a left endpoint, insert this point into the active line list, where each line is represented by its left endpoint in the list and a line is in the list if it contains current scan point.
 2. if current point is a right endpoint, scan the active line list and do the following operations:
 - i. if current left endpoint is its left endpoint, remove this left endpoint from the active line list.
 - ii. if current left endpoint is not its left endpoint, compute the overlapping length of these two lines.
- (b) (100 pts) Implement this method and that of Problem 1(c), and compare their runtimes. Write a note to analyze these two methods to explain why this or that one is faster from your viewpoint. Submit your note and two programs to E3. (due in 12/13)
- (c) (additional 10 pts) If you are not satisfied with the performance of method mentioned in Problem 2.a, do you have any refined method to improve its performance (still use array structure)? Submit your refined program and a note to show your refined method.

2. Bonus

As we declare an array in a C program, we may find that the array size cannot be too large. Try to declare an array whose size is more than 10^6 in the way like "int ary[1000000];". If it cannot work, find a way to declare an array whose size is larger than 10^6 and explain why traditional way does not work and your way works. It's due in the midnight of today.