

CSC265 Fall 2021 Homework Assignment 1

due Wednesday September 22, 2021

A line segment s on the plane can be represented by four numbers, its slope $s.m$, the x coordinate $s.l$ of its left end point, the x coordinate $s.r$ of its right end point, and the y intercept $s.b$ of the line containing s , provided s is not vertical. A vertical line segment s has slope $s.m = \infty$. It is represented by the y coordinate $s.l$ of its lower end point, the y coordinate $s.r$ of its upper end point, and the x intercept $s.b$ of the line containing s .

A set of line segments S is *disjoint* if no two line segments in S intersect one another.

In this assignment, you will design algorithms that, given two sets of disjoint line segments, S and S' , counts the number of (unordered) pairs of lines in $S \cup S'$ that intersect. For example, if S is the set of two black lines and S' is the set of three red lines in Figure 1, then there are 3 intersecting pairs of lines.

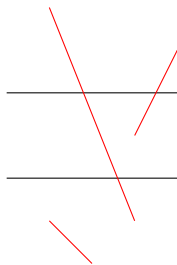


Figure 1: The intersections between two sets of disjoint line segments

1. Suppose that $s.l = 0$ and $s.r = 1$ for each line segment $s \in S \cup S'$, the line segments in $S = \{s_1, \dots, s_n\}$ are sorted in increasing order by their y intercepts (i.e. $s_1.b < s_2.b < \dots < s_n.b$), and the line segments in $S' = \{s'_1, \dots, s'_{n'}\}$ are sorted in increasing order by their y intercepts (i.e. $s'_1.b < s'_2.b < \dots < s'_{n'}.b$). Give an algorithm (in pseudocode) that counts the number of pairs of lines in $S \cup S'$ that intersect. Your algorithm must run in $O(n + n')$ time, where $n = |S|$ and $n' = |S'|$.

First, briefly explain, at a high level, how your algorithm works. Then give the pseudocode. Your explanation should NOT be a line by line description of the pseudocode.

Explain why your algorithm runs in the required time.

2. Prove that your algorithm in part 1 is correct. If your code is iterative, carefully state relevant loop invariants and prove that they are invariants by induction. If your code is recursive, carefully state relevant properties of your algorithm and prove them by induction.
3. Suppose that, as in part 1, $s'.l = 0$ and $s'.r = 1$ for each line segment $s' \in S'$, and the line segments in $S' = \{s'_1, \dots, s'_{n'}\}$ are sorted in increasing order by their y intercepts (i.e. $s'_1.b < s'_2.b < \dots < s'_{n'}.b$). However, $0 \leq s.l \leq s.r \leq 1$ for each non-vertical line segment $s \in S$ and $0 \leq s.b \leq 1$ for each vertical line segment $s \in S$. Give an algorithm (in pseudocode) for determining the number of intersections between S and S' . Your algorithm must run in $O(n \log n')$ time, where $n = |S|$ and $n' = |S'|$.

First, briefly explain, at a high level, how your algorithm works. Then give the pseudocode. Your explanation should NOT be a line by line description of the pseudocode.

Explain why your algorithm runs in the required time.

4. Prove that your algorithm in part 3 is correct. If your code is iterative, carefully state relevant loop invariants and prove that they are invariants by induction. If your code is recursive, carefully state relevant properties of your algorithm and prove them by induction.