

I collaborated with:

Problem

Solve Problem 5 from Chapter 5 of your text. Divide and conquer works well here but there are other methods of solution; feel free to try a different way if D-C isn't working for you!

Hint: What does the structure of the set of *all* uppermost points for a set of lines L look like? If you had these structures for sets L_1 and L_2 of lines, could you compute the structure for $L_1 \cup L_2$?

Solution

Sort the lines by increasing slopes. Our base case is when there are two intersecting lines. We know that the line with the lowest slope is visible to the left of the intersection and the line with the greater slope is visible to the right of the intersection. We must also keep track of where the lines intersect which can be done in $O(1)$. If we take a divide and conquer approach we now have two visible segments i and j of lines we must integrate. We know where the lines of i and j intersect so we can form line segments for all lines. Take the first line segment of i and compare it to the first line segment of j only if these segments share an x-coordinate. If these segments do not intersect, then we take the uppermost one, and move on to the next segment of the set we did not select. If they do intersect, then we take the approach of our base case (the line segment with the lowest slope is visible to the left of the intersection and the line segment with the greater slope is visible to the right of the intersection). If our line segments m and n do not share an x-coordinate then we move on to the next line segment of the set containing the leftmost segment, m or n .

Recurrence relation: $T(n) = 2T(n/2) + cn = O(n \log n)$ as walking through our i and j sets is a linear operation.

Proof. Base Case: Let the number of lines n be two. Since they intersect, the line with the lowest slope is visible to the left of the intersection and the line with the greater slope is visible to the right of the intersection. We can simply return the visible line segments to make the algorithm runs faster.

Inductive Hypothesis: Assume our algorithm works for i number of lines where $2 \leq i \leq k$ for some integer k .

Inductive Step: Let L be a set of $k + 1$ number of lines. We must show that our algorithm returns the visible lines of L . Our divide and conquer approach assures we have two smaller subproblems and our inductive hypothesis allows us to claim that these subproblems are correct as those sets are of size less than $k + 1$. Let these two sets of line be I and J . I and J are sets of line segments visible within their own sets. If we take the leftmost segments from I and J s.t. they share an x-coordinate, when we simply take the uppermost one. These sets are sorted in increasing order of slopes, so finding the next segment in a set takes constant time. After every step of our loop, we select the uppermost segment, checking intersections, line segments that do not share an x-coordinate, and segments that do not intersect. \square