GeeksforGeeks

A computer science portal for geeks

GeeksQuiz

Given n line segments, find if any two segments intersect

We have discussed the problem to detect if two given line segments intersect or not. In this post, we extend the problem. Here we are given n line segments and we need to find out if any two line segments intersect or not.

Naive Algorithm A naive solution to solve this problem is to check every pair of lines and check if the pair intersects or not. We can check two line segments in O(1) time. Therefore, this approach takes $O(n^2)$.

Sweep Line Algorithm: We can solve this problem in **O(nLogn)** time using Sweep Line Algorithm. The algorithm first sorts the end points along the x axis from left to right, then it passes a vertical line through all points from left to right and checks for intersections. Following are detailed steps.

1) Let there be n given lines. There must be 2n end points to represent the n lines. Sort all points according to x coordinates. While sorting maintain a flag to indicate whether this point is left point of its line or right point.

2) Start from the leftmost point. Do following for every point
…..**a)** If the current point is a left point of its line segment, check for intersection of its line segment with the segments just above and below it. And add its line to *active* line segments (line segments for which left end point is seen, but right end point is not seen yet). Note that we consider only those neighbors which are still active.
….**b)** If the current point is a right point, remove its line segment from active list and check whether its two active neighbors (points just above and below) intersect with each other.

The step 2 is like passing a vertical line from all points starting from the leftmost point to the rightmost point. That is why this algorithm is called Sweep Line Algorithm. The Sweep Line technique is useful in many other geometric algorithms like calculating the 2D Voronoi diagram

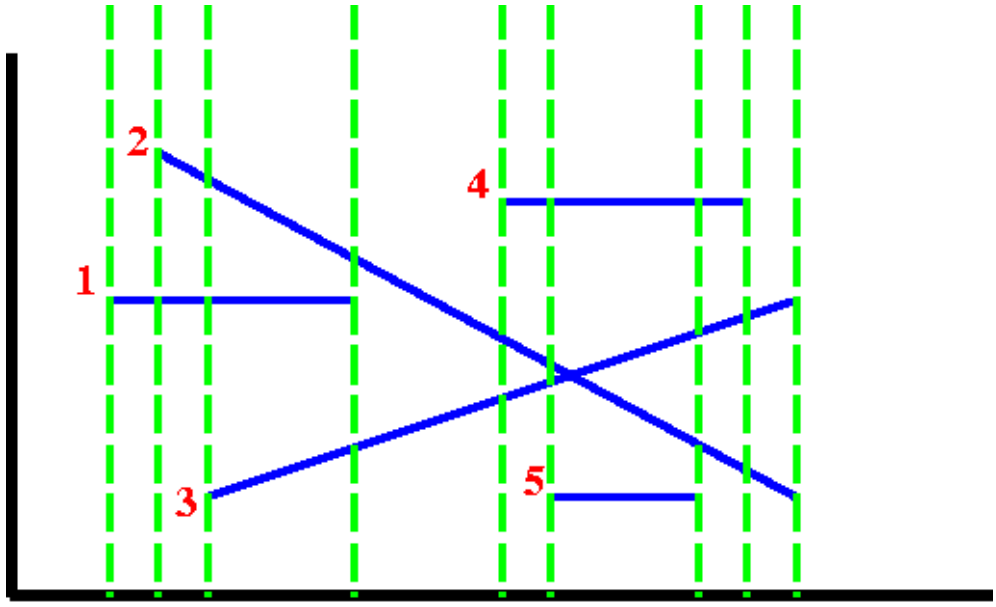
What data structures should be used for efficient implementation?
In step 2, we need to store all active line segments. We need to do following operations efficiently:
a) Insert a new line segment
b) Delete a line segment
c) Find predecessor and successor according to y coordinate values
The obvious choice for above operations is Self-Balancing Binary Search Tree like AVL Tree, Red Black Tree. With a Self-Balancing BST, we can do all of the above operations in O(Logn) time.
Also, in step 1, instead of sorting, we can use min heap data structure. Building a min heap takes O(n) time and every extract min operation takes O(Logn) time (See this).

PseudoCode:
The following pseudocode doesn't use heap. It simply sort the array.

```
sweepLineIntersection(Points[0..2n-1]):
1. Sort Points[] from left to right (according to x coordinate)

2. Create an empty Self-Balancing BST T. It will contain all active line
   Segments ordered by y coordinate.

// Process all 2n points
3. for i = 0 to 2n-1

    // If this point is left end of its line
    if (Points[i].isLeft)
       T.insert(Points[i].line())  // Insert into the tree

       // Check if this points intersects with its predecessor and successor
       if ( doIntersect(Points[i].line(), T.pred(Points[i].line()) )
         return true
       if ( doIntersect(Points[i].line(), T.succ(Points[i].line()) )
         return true

    else  // If it's a right end of its line
       // Check if its predecessor and successor intersect with each other
       if ( doIntersect(T.pred(Points[i].line(), T.succ(Points[i].line()))
         return true
       T.delete(Points[i].line())  // Delete from tree

4. return False
```

Example:

Let us consider the following example taken from here. There are 5 line segments 1, 2, 3, 4 and 5. The dotted green lines show sweep lines.



sweep lines

Following are steps followed by the algorithm. All points from left to right are processed one by one. We maintain a self-balancing binary search tree.

Left end point of line segment 1 is processed: 1 is inserted into the Tree. The tree contains 1. No intersection.

Left end point of line segment 2 is processed: Intersection of 1 and 2 is checked. 2 is inserted into the Tree. No intersection. The tree contains 1, 2.

Left end point of line segment 3 is processed: Intersection of 3 with 1 is checked. No intersection. 3 is inserted into the Tree. The tree contains 2, 1, 3.

Right end point of line segment 1 is processed: 1 is deleted from the Tree. Intersection of 2 and 3 is checked. Intersection of 2 and 3 is reported. The tree contains 2, 3. Note that the above pseudocode returns at this point. We can continue from here to report all intersection points.

Left end point of line segment 4 is processed: Intersections of line 4 with lines 2 and 3 are checked. No intersection. 4 is inserted into the Tree. The tree contains 2, 4, 3.

Left end point of line segment 5 is processed: Intersection of 5 with 3 is checked. No intersection. 4 is inserted into the Tree. The tree contains 2, 4, 3, 5.

Right end point of line segment 5 is processed: 5 is deleted from the Tree. The tree contains 2, 4, 3.

Right end point of line segment 4 is processed: 4 is deleted from the Tree. The tree contains 2, 4, 3. Intersection of 2 with 3 is checked. Intersection of 2 with 3 is reported. The tree contains 2, 3. Note that the intersection of 2 and 3 is reported again. We can add some logic to check for duplicates.

Right end point of line segment 2 and 3 are processed: Both are deleted from tree and tree becomes

empty.

Time Complexity: The first step is sorting which takes O(nLogn) time. The second step process 2n points and for processing every point, it takes O(Logn) time. Therefore, overall time complexity is O(nLogn)

References:
http://www.cs.uiuc.edu/~jeffe/teaching/373/notes/x06-sweepline.pdf
http://courses.csail.mit.edu/6.006/spring11/lectures/lec24.pdf
http://www.youtube.com/watch?v=dePDHVovJlE
http://www.eecs.wsu.edu/~cook/aa/lectures/l25/node10.html

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Related Topics:

- Linearity of Expectation
- Iterative Tower of Hanoi
- Count possible ways to construct buildings
- Build Lowest Number by Removing n digits from a given number
- Set Cover Problem | Set 1 (Greedy Approximate Algorithm)
- Find number of days between two given dates
- How to print maximum number of A's using given four keys
- Write an iterative O(Log y) function for pow(x, y)

Tags: geometric algorithms

Writing code in comment? Please use **ideone.com** and share the link here.

Abhishek · 3 months ago

Looks like the algo is wrong in below step:

Right end point of line segment 5 is processed: 5 is deleted from the Tree. The tree contains 2,4, 3.

What if 5 was a little longer and was intersecting 2, where are we checking for that?

#WrongAlgo

• Reply • Share ›

Darya Prokurat • 7 months ago

It is not obvious for me, that it is enough to make only these checking to find ANY ONE intersect, but I am fully confident that it is not enough to find ALL of them. On adding new end point, we make 2 checks, and 1 check after deleting: so not more then 3 check per line in total = 3*N, and some of them can be duplicates. But we can draw N lines so, that all of then will be intersecting: N*(N-1)/2 pairs. For N > 7, intersections more than we make checks.
This is regarding to note: "Note that the above pseudocode returns at this point. We can continue from here to report all intersection points".

Please, let me know if I missed something.

• Reply • Share ›

Alfred • 8 months ago

If the points is like this, I think the alg is not correct.

Points: 1: {(0, 0), (0, 4)},

2: {(1, 1), (5, 7)},

3: {(2, 2), (3, -1)}

The Order is :

1. Insert 1 left

Tree: 1

2, Insert 2 left

Tree: 2, 1

check (2, 1)

3, Insert 3 left

see more

• Reply • Share ›

Tejas Patel • a year ago

Why do we need to check only two segments and not more?

• Reply • Share ›

Savan Popat • a year ago

In "Left end point of line segment 5 is processed" , it should be "5 inserted in to the tree" instead of 4.

• Reply • Share ›

annonymoe • a year ago

I am new here but Can we not do something like this
We use result from 1st line segment comparison with other line segments and divide the lines into two buckets one to the left of line segment and 1 to the right based on the orientation properties

Reply • Share ›

Kartik annonymoe • a year ago

Divide and Conquer is good idea. But the question is, how to write the combine/conquer step, it should be O(n). For example, for closest pair of points problem (http://www.geeksforgeeks.org/c..., there is a combine step that takes linear time. Can there be something similar here?

Reply • Share ›

Vivek VV • a year ago

I had one doubt regarding the method. In step 1/2 how you check for the line segment/points just above and below the point in the program. Thanks in advance.

Reply • Share ›

GeeksforGeeks Mod Vivek VV • a year ago

In BST, we keep active line segments ordered according to their y coordinates. To find the point just below, we need predecessor in BST. Similarly to find the successor, we need successor in BST.

2 Reply • Share ›

Vivek VV GeeksforGeeks • a year ago

Thanks for the quick reply. I am still not clear :(. When we examine the left point for 3rd line, why we check the intersection with line 2. We only have successor of line 3 as line 1. But left point of line 2 is not the predecessor of line 3, even by y axis.

2 Reply • Share ›

GeeksforGeeks Mod Vivek VV • a year ago

Vivek, Thanks for pointing this out, intersection of 3 should only be checked with 1. We have updated the example.

Reply • Share ›

Deepak GeeksforGeeks • a year ago

Following up on this, which y-coordinate do we use to order the points in a bst? The line segment has two y-coordinates right. Do we use the left one?

Reply • Share ›

Deepak ➚ Deepak • a year ago

And if we use the left one, why?

• Reply • Share ›

vivek • a year ago

There is a typo in example dry run.
Following is repeated.
Left end point of line segment 2 is processed:

It should be

Left end point of line segment 3 is processed:

• Reply • Share ›

GeeksforGeeks Mod ➚ vivek • a year ago

Thanks for pointing this out. We have updated the example.

• Reply • Share ›

xxmajia • a year ago

"Building a min heap takes O(n)"

should be O(nlgn), maintain a heap will cause O(lgn), for building a heap, you need to maintain it n times, hence O(nlgn)

2 • Reply • Share ›

GeeksforGeeks Mod ➚ xxmajia • a year ago

The following wiki link also has proof.

http://en.wikipedia.org/wiki/B...

1 • Reply • Share ›

GeeksforGeeks Mod ➚ xxmajia • a year ago

Building a min heap from given n elements takes O(n) time. Please refer the CLRS book for proof.

1 • Reply • Share ›

xxmajia ➚ GeeksforGeeks • a year ago

my bad, it is O(n), thanks for correcting me

2 • Reply • Share ›

- Interview Experiences
- Advanced Data Structures
- Dynamic Programming
- Greedy Algorithms
- Backtracking
- Pattern Searching
- Divide & Conquer
- Mathematical Algorithms
- Recursion
- Geometric Algorithms

Popular Posts

- All permutations of a given string
- Memory Layout of C Programs
- Understanding “extern” keyword in C
- Median of two sorted arrays
- Tree traversal without recursion and without stack!
- Structure Member Alignment, Padding and Data Packing
- Intersection point of two Linked Lists
- Lowest Common Ancestor in a BST.
- Check if a binary tree is BST or not
- Sorted Linked List to Balanced BST

Recent Comments

- radek

  hey venki..is there a way to implement the same...

  The Ubiquitous Binary Search | Set 1 · 1 minute ago

- radek

  hey..is there a way to implement the same...

  The Ubiquitous Binary Search | Set 1 · 2 minutes ago

- Nikhil kumar

  public class...

  Print missing elements that lie in range 0 – 99 · 35 minutes ago

- Ashish Aggarwal

  Try Data Structures and Algorithms Made Easy -...

  Algorithms · 57 minutes ago

- Vlad

  Thanks. Very interesting lectures.

  Expected Number of Trials until Success · 2 hours ago

- cfh

  My implementation which prints the index of the...

  Longest Even Length Substring such that Sum of First and Second Half is same · 2 hours ago