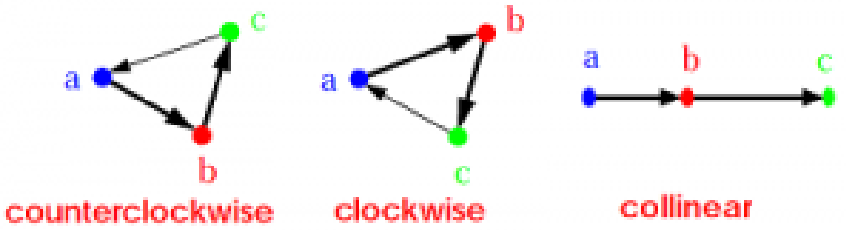
How to check if two given line segments intersect?

Given two line segments (p1, q1) and (p2, q2), find if the given line segments intersect with each other.

Before we discuss solution, let us define notion of **orientation**. Orientation of an ordered triplet of points in the plane can be
–counterclockwise
–clockwise
–colinear
The following diagram shows different possible orientations of (a, b, c)



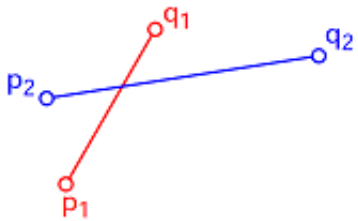
Note the word 'ordered' here. Orientation of (a, b, c) may be different from orientation of (c, b, a).

How is Orientation useful here?
Two segments (p1,q1) and (p2,q2) intersect if and only if one of the following two conditions is verified

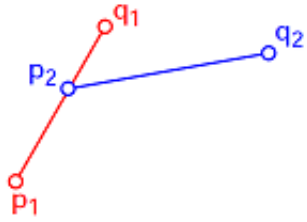
1. ***General Case:***
– (p1, q1, p2) and (p1, q1, q2) have different orientations and
– (p2, q2, p1) and (p2, q2, q1) have different orientations

2. ***Special Case***
– (p1, q1, p2), (p1, q1, q2), (p2, q2, p1), and (p2, q2, q1) are all collinear and
– the x-projections of (p1, q1) and (p2, q2) intersect
– the y-projections of (p1, q1) and (p2, q2) intersect

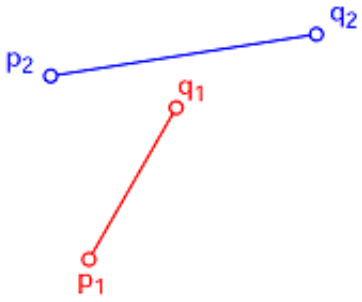
Examples of General Case:



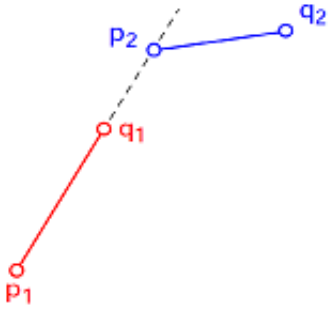
Example 1: Orientations of (p1, q1, p2) and (p1, q1, q2) are different. Orientations of (p2, q2, p1) and (p2, q2, q1) are also different



Example 2: Orientations of (p1, q1, p2) and (p1, q1, q2) are different. Orientations of (p2, q2, p1) and (p2, q2, q1) are also different



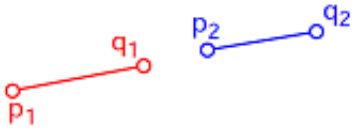
Example 3: Orientations of (p1, q1, p2) and (p1, q1, q2) are different. Orientations of (p2, q2, p1) and (p2, q2, q1) are **same**



Example 4: Orientations of (p1, q1, p2) and (p1, q1, q2) are different. Orientations of (p2, q2, p1) and (p2, q2, q1) are **same**

Examples of Special Case:

Example 1: All points are colinear. The x-projections of (p1, q1) and (p2, q2) intersect. The y-projections of (p1, q1) and (p2, q2) intersect



Example 2: All points are colinear. The x-projections of (p1, q1) and (p2, q2) do not intersect. The y-projections of (p1, q1) and do not (p2, q2) intersect

Following is C++ implementation based on above idea.

```
// A C++ program to check if two given line segments intersect
#include <iostream>
using namespace std;

struct Point
{
    int x;
    int y;
};

// Given three colinear points p, q, r, the function checks if
// point q lies on line segment 'pr'
bool onSegment(Point p, Point q, Point r)
{
    if (q.x <= max(p.x, r.x) && q.x >= min(p.x, r.x) &&
        q.y <= max(p.y, r.y) && q.y >= min(p.y, r.y))
       return true;

    return false;
}

// To find orientation of ordered triplet (p, q, r).
// The function returns following values
// 0 --> p, q and r are colinear
// 1 --> Clockwise
// 2 --> Counterclockwise
int orientation(Point p, Point q, Point r)
{
    // See 10th slides from following link for derivation of the formula
    // http://www.dcs.gla.ac.uk/~pat/52233/slides/Geometry1x1.pdf
    int val = (q.y - p.y) * (r.x - q.x) -
              (q.x - p.x) * (r.y - q.y);

    if (val == 0) return 0;  // colinear

    return (val > 0)? 1: 2; // clock or counterclock wise
}

// The main function that returns true if line segment 'p1q1'
// and 'p2q2' intersect.
bool doIntersect(Point p1, Point q1, Point p2, Point q2)
{
    // Find the four orientations needed for general and
```

```
    // special cases
    int o1 = orientation(p1, q1, p2);
    int o2 = orientation(p1, q1, q2);
    int o3 = orientation(p2, q2, p1);
    int o4 = orientation(p2, q2, q1);

    // General case
    if (o1 != o2 && o3 != o4)
        return true;

    // Special Cases
    // p1, q1 and p2 are colinear and p2 lies on segment p1q1
    if (o1 == 0 && onSegment(p1, p2, q1)) return true;

    // p1, q1 and p2 are colinear and q2 lies on segment p1q1
    if (o2 == 0 && onSegment(p1, q2, q1)) return true;

    // p2, q2 and p1 are colinear and p1 lies on segment p2q2
    if (o3 == 0 && onSegment(p2, p1, q2)) return true;

     // p2, q2 and q1 are colinear and q1 lies on segment p2q2
    if (o4 == 0 && onSegment(p2, q1, q2)) return true;

    return false; // Doesn't fall in any of the above cases
}

// Driver program to test above functions
int main()
{
    struct Point p1 = {1, 1}, q1 = {10, 1};
    struct Point p2 = {1, 2}, q2 = {10, 2};

    doIntersect(p1, q1, p2, q2)? cout << "Yes\n": cout << "No\n";

    p1 = {10, 0}, q1 = {0, 10};
    p2 = {0, 0}, q2 = {10, 10};
    doIntersect(p1, q1, p2, q2)? cout << "Yes\n": cout << "No\n";

    p1 = {-5, -5}, q1 = {0, 0};
    p2 = {1, 1}, q2 = {10, 10};
    doIntersect(p1, q1, p2, q2)? cout << "Yes\n": cout << "No\n";

    return 0;
}
```

Output:

```
No
Yes
No
```

Sources:
http://www.dcs.gla.ac.uk/~pat/52233/slides/Geometry1x1.pdf
Introduction to Algorithms 3rd Edition by Clifford Stein, Thomas H. Cormen, Charles E. Leiserson,

Ronald L. Rivest

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, MathematicalAlgo

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

Prasenjit Mondal • a month ago

onSegment implementation is wrong. It considers parallel lines also. It would return true in case of a point outside and close to the line.

The point is on the line segment or not can be checked using several methods like following:
1) Using line equation
2) Using linear interpolation technique
3) Using dot product
4) Calculating length of the two segments and sum must be equal to total length

Admin please correct the post.

Amar Vashishth Prasenjit Mondal • 10 days ago

no, its correct

gaurav Prasenjit Mondal • 15 days ago

we are going for onSegment(..) checking only if points are co-linear.
End points of two parallel line segments are not co-linear . Hence , we are checking for a valid co-linear case.

Reply • Share ›

Abhishek • 3 months ago

How does below code confirms whether a point falls on a line segment?

// Given three colinear points p, q, r, the function checks if

// point q lies on line segment 'pr'

bool onSegment(Point p, Point q, Point r)

{

if (q.x <= max(p.x, r.x) && q.x >= min(p.x, r.x) &&

q.y <= max(p.y, r.y) && q.y >= min(p.y, r.y))

return true;

return false;

}

It would say true, for points which are slightly not on the line as well. It is just checking whether q has x and y in between of x's and y's of p and r.

1 Reply • Share ›

sandeep • 5 months ago

i meant to say the best

Reply • Share ›

Morgan sandeep • 3 months ago

good post, clear enough for me to understand!!

Reply • Share ›

sandeep Morgan • 3 months ago

so what? why r u telling me?

Reply • Share ›

sandeep • 5 months ago

this is the worst geeksforgeeks post I've seen so far

4 Reply • Share ›

Rajarshee Mitra • 5 months ago

Two line segments will intersect only if they have different slopes. So, given (p1,q1) and (p2,q2) as two lines where p1=(xp1,yp1);q1=(xq1,yq1); p2=(xp2,yp2) and q2=(xq2,yq2) it is suffice to check whether (yq2-yp2)/(xq2-xp2) == (yq1-yp1)/(xq1-xp1). If equal, they don't intersect , else they do.

Reply • Share ›

mike ➚ Rajarshee Mitra • 4 months ago

they are line segments, not lines.

2 Reply • Share ›

Deepesh Maheshwari • 5 months ago

Hi, above algo fails for this case.
{10, 0},{10, 10},{5,5},{12344,5}
Here, when we compute orientation for following case :
orienation({10, 0},{10, 10},{5,5})
orienation({10, 0},{10, 10},{12344,5})

This is giving same orientation.
Please explain how to resolve this bug.

Reply • Share ›

Harry Potter • 8 months ago

Is there any other way for the orientation to be 0 ? If not then why are we checking the orientation==0 and then again checking that the the point lies on the segment ?

Reply • Share ›

Sandu Robert ➚ Harry Potter • 6 months ago

orientation is 0 if p,q,r are colinear but we can have p1-p2-q1 or p2-p1-q1 or p1-q1-p2 ....

Reply • Share ›

ali • 8 months ago

Hi
this code is good.
but I'm looking for this code in matlab.
is there any one help me.time is running out. and I don't do anything.
I have to find the coordinate of intersection of all lines in a chess board image that find them by hough transform.
my email address is: adelanteh1365@yahoo.com
thanks in advance.

Reply • Share ›

Bijoy Singh • 9 months ago

Cant we do something of this sort: Given the 2 points on the line segments, determine the 2

Cant we do something of this sort. Given the 2 points on the line segments, determine the 2 lines. Find their intersection point , say P(x,y). Now find the ratio P divides both of the line segment. If this ratio is in 0->1 in both the lines this means the points in inside the line segments, hence they intersect.

• Reply • Share ›

sugarcane_farmer • a year ago

I think this can be better solved using geometry and vector knowledge. See this for more: http://community.topcoder.com/...

1 • Reply • Share ›

Guest • a year ago

I don't think this implementation distinguishes between "collinear" and "parallel".
Compare two lines, (10,10) - (20,20) against (11,11) - (21,21) - OnSegment returns TRUE, when it isn't.

• Reply • Share ›

lkreinitz ➚ Guest • 10 months ago

the orientation conditions preclude this case

• Reply • Share ›

Kaidul Islam Sazal • a year ago

Is it possible in this code to determine in which point the lines intersect?

• Reply • Share ›

Amit • a year ago

onSsegment should be

if (p.x * (q.y - r.y) + q.x * (r.y - p.y) + r.x*(p.y - q.y) == 0)
return true;

• Reply • Share ›

Fredrik • 2 years ago

The code seems to be optimized for the case where the segments do intersect. Any suggestions for optimizing the code for the case where the segments probably do not intersect?

• Reply • Share ›

clotho • 2 years ago

An easier way:
Segment L1 has edges A=(a1,a2), A'=(a1',a2').
Segment L2 has edges B=(b1,b2), B'=(b1',b2').
Segment L1 is the set of points tA'+(1-t)A, where 0<=t<=1.
Segment L2 is the set of points sB'+(1-s)B, where 0<=s<=1.

Segment L1 meet segment L2 if and only if for some t and s we have
tA'+(1-t)A=sB'+(1-s)B
The solution of this with respect to t and s is

t=((-b?'a?+b?'b?+b?a?+a?b?'-a?b?-b?b?')/(b?'a?'-b?'a?-b?a?'+b?a?-a?'b?'+a?'b?+a?b?'-a?b?))

s=((-a?b?+a?'b?-a?a?'+b?a?+a?'a?-b?a?')/(b?'a?'-b?'a?-b?a?'+b?a?-a?'b??+a?'b?+a?b?'-a?b?))

So check if the above two numbers are both >=0 and <=1.
:)

• Reply • Share ›

Mayur Shah • 2 years ago

You can check it by using slope of lines whether the two lines are intersecting or not. For example two line with same slopes will never intersect at all and vice-versa!

• Reply • Share ›

GradLifeWoes ➚ Mayur Shah • a year ago

That would be the case if they are lines and not line segments.
Even in the case of lines another check has to be dome to determine that the two lines are not same.
So we should check they are not collinear and have same slopes to determine that they never intersect.

2 • Reply • Share ›

sarat G • 2 years ago

Hey,

What's the point in doing all this stuff...if the slopes of two line segments are equal we can they never intersect...in any other case the two line segments always intersects..

```
/* Paste your code here (You may delete these lines if not writing code) */
```

• Reply • Share ›

kartik ➚ sarat G • 2 years ago

That is why the topic says line *Segments* :)

7 • Reply • Share ›

Mohan • 2 years ago

I appreciate the efforts taken by geeksforgeeks community in posting useful problems and their solutions here but this one was very poorly explained.

• Reply • Share ›

??? • 2 years ago

the pdf is misleading, what he needs to say is : vector product.
check this wiki will explain everything:
http://en.wikipedia.org/wiki/C...

• Reply • Share ›

xxmajia • 2 years ago

For 10th page of http://www.dcs.gla.ac.uk/~pat/...

i don't get it, its true only when p, q, r 's x-projection are increasing

can anyone explain it a little bit?

```
/* Paste your code here (You may delete these lines if not writing code) */
```

• Reply • Share ›

kartik ↗ xxmajia • 2 years ago

I think CLRS book would be helpful for better understanding.

• Reply • Share ›

IsAs • 2 years ago

distance(P1, Q1) = distance between P1 and Q1.
If point K lies on a line segment then following equation is true : distance(P1,Q1) = distance(P1,K) + distance(K,Q1)

Find the slopes of two line segments and let's say if they are not equal - then

1) Form two equations of the form ax+by+c = 0
2) Compute the intersecting point by solving above line equations.Let's call our intersecting point as I.
3) Verify whether distance(P1, Q1) == distance(P1,I) + distance(I,Q1)
4) Verify the same with P2 and Q2 as well
5) If intersecting point lies on both the line segments then above two verifications should succeed

If slopes are equal then check if their equations are same or not. If equations are different then they are parallel lines - no intersecting point

If equations are same then the points are collinear - verify the following (one of the points of the line segment should lie on the other)

distance(P1,Q1) == distance(P1,P2) + distance(P2, Q1)
distance(P1,Q1) == distance(P1,Q2) + distance(Q2, Q1)

If any of the above statements are true then the line segments intersect

3 ^ | v • Reply • Share ›

blackball • 2 years ago

How about this one:

```
/* http://stackoverflow.com/questions/563198/how-do-you-detect-where-two-line-segments-int

struct point {
    float x, y;
};

static inline int
line_intersect_2d(struct point a, struct point b,
                  struct point c, struct point d) {
    struct point cmp, r, s;
    cmp.x = c.x - a.x;
    cmp.y = c.y - a.y;
    r.x = b.x - a.x;
    r.y = b.y - a.y;
    s.x = d.x - c.x;
    s.y = d.y - c.y;
```

see more

^ | v • Reply • Share ›

Dixit Sethi • 2 years ago

Using basic maths, this can be done easily.

First take the two points of first line. say (x11, y11) and (x12, y12). This is line segment and we can find the equation of line passing through these points.

eqn: "y*(x12-x11)- x*(y12-y11) + x11(y12-y11) - y11(x12-x11)".

This eqn will equate to zero if x and y are points lying anywhere on the line (not just the line segment). If the point (x,y) is above the line, then eqn value is positive and if the point is below the line, then it is negative.

For cases like example 1, we need the two endpoints of the second line on the opposite sides of the first line and vice versa. Check the value of the equation of first line for the two (x,y) end points of the second line. The two eqn values must be opposite in sign. Similarly check the two endpoints of the first line with the eqn of second line. These eqn values must also be opposite in sign. If thats the case, line segments are intersecting. Example 3 will fail in this case, stating it non intersecting.

Now, the case of both the endpoints of one line segment lying on the line of other line segment

Now, the case of both the endpoints of one line segment lying on the line of other line segment

see more

1 • Reply • Share ›

Parin • 2 years ago

How is orientation calculated?

1 • Reply • Share ›

GeeksforGeeks ➚ Parin • 2 years ago

Please see slide 10 of the following link

http://www.dcs.gla.ac.uk/~pat/...

1 • Reply • Share ›

Parin ➚ GeeksforGeeks • 2 years ago

Suppose the points are P1(-5,-5) P2(0,0) P3(-4,-3).
Then this is counterclockwise.But, here slope of p2p3 won't be greater then slope of p1p2.

• Reply • Share ›

kartik ➚ Parin • 2 years ago

The program seems to be printing counterclockwise only. See http://ideone.com/6DuGJ5

• Reply • Share ›

Ravi Kesh Singh • 2 years ago

Another solution can be as following.

lets say
struct lineEqu{
int x;
int y;
int c; //line equ y-mx-c=0 here x =-m;y=1;z=-c.
}

populate lineEqu for(p1, q1) and (p2, q2) and write a method which return intersecting point (x, y) if (x, y) lies between (p1, q1) as well as (p2, q2) then they will intersect, if intersection point exist and not lies in either (p1, q1) or (p2, q2) then solution does not exist.

If intersection point does not exist then also sol does not exist.

• Reply • Share ›

Verma Shailendra • 2 years ago

agree it will always detect the intersection. but may give false result while not intersecting.

• Reply • Share ›

Hitesh • 2 years ago

If you want to apply your basic math skills, then solve an interesting problem called Convex-Hull Problem.

In which you would find some interesting small problems like:

- Check whether the particular point lies inside the given polygon
- Find an angle between the 3 input points
- Given the input points, rearrange them in specific order to form the largest possible polygon (in terms of its area)
- Solve the 2 lines for possible intersection

• Reply • Share ›

GeeksforGeeks ➚ Hitesh • 2 years ago

@Hitesh: Thanks for sharing your inputs.
We have published an article on point and polygon problem.
We will soon be covering Convex Hull problem also.

You can also publish an article by mailing it to contribute@geeksforgeeks.org.

Keep it up!

• Reply • Share ›

Uddhav • 2 years ago

I did not understand this condition " q.y = min(r.y, r.y) " ?
Why use max and min functions between same two values ??

Thank you

• Reply • Share ›

Hitesh • 2 years ago

```
class Point {

        private double x;
        private double y;

        public Point( ) {

        }

        public Point( double x, double y ) {

                this.x = x;
```

```
            this.y = y;

        }

        public double getX( ) {
```

see more

• Reply • Share ›

Sanjay Ahuja • 2 years ago

This approach will work but you may be missing one case when lines are parallel. In that case x projection and y projection may still intersect but lines are not intersecting.

• Reply • Share ›

Pratik • 2 years ago

Here orientation is nothing but the cross product.
Orientation (p,q,r) is cross product of Vector(p to q) and Vector(p to r).

But yeah. Good code!!

• Reply • Share ›

timus ➚ Pratik • 2 years ago

THANKs, this really helped....

• Reply • Share ›

Abhilash Reddy • 2 years ago

in if statement you can simply put.
q.x == max(p.x, r.x) && q.y == max(r.y, r.y).

i don&#039t know about the implementation I was just curious to know code optimization and can some one explain in more simplar way.

• Reply • Share ›

abhilash • 2 years ago

in frst if statement what is the point of keeping
q.x = min(p.x, r.x) &&
q.y = min(r.y, r.y)

you can simply keep q.x==max(p.x,r.x)&&q.y==min(r.y,r.y)

i dont know about the implementation of the program i was just curious to know in code optimization

• Reply • Share ›

Verma Shailendra • 2 years ago

I feel better and simpler solutions is we can find out only on the basis of projection on x and y axis. (no need of orientation at all).
ie (P1, Q1) and (P2, Q2) &#039s x projection (px1, qx1) will line on same side of (px2, qx2) it means x projection do not intersect, so if x and y both projection do not intersect than line segment do not intersect.

1 ^ | v • Reply • Share ›

Yelnil Gabo • 2 years ago

I like the colors. :P

^ | v • 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

Follow @GeeksforGeeks

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

-