

# GeeksforGeeks

A computer science portal for geeks

## GeeksQuiz

- [Home](#)
- [Algorithms](#)
- [DS](#)
- [GATE](#)
- [Interview Corner](#)
- [Q&A](#)
- [C](#)
- [C++](#)
- [Java](#)
- [Books](#)
- [Contribute](#)
- [Ask a Q](#)
- [About](#)

[Array](#)

[Bit Magic](#)

[C/C++](#)

[Articles](#)

[GFacts](#)

[Linked List](#)

[MCQ](#)

[Misc](#)

[Output](#)

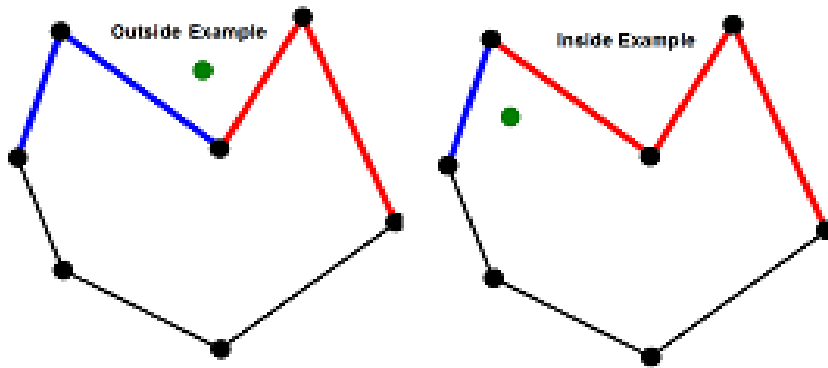
[String](#)

[Tree](#)

[Graph](#)

## How to check if a given point lies inside or outside a polygon?

Given a polygon and a point 'p', find if 'p' lies inside the polygon or not. The points lying on the border are considered inside.

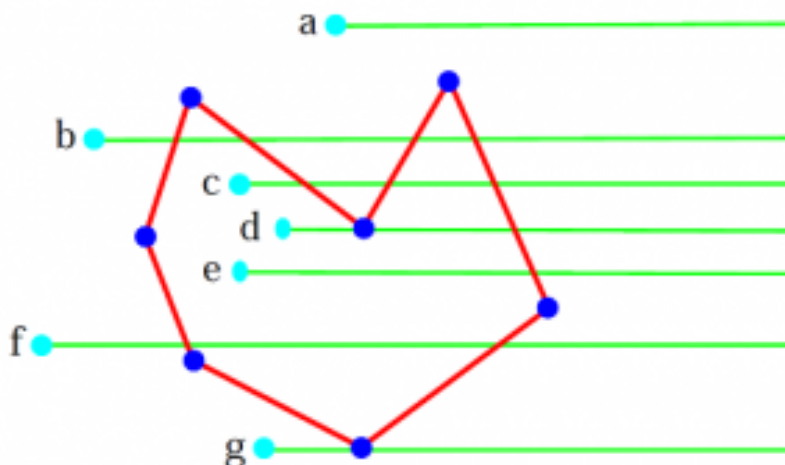


We strongly recommend to see the following post first.

[How to check if two given line segments intersect?](#)

Following is a simple idea to check whether a point is inside or outside.

- 1) Draw a horizontal line to the right of each point and extend it to infinity
- 1) Count the number of times the line intersects with polygon edges.
- 2) A point is inside the polygon if either count of intersections is odd or point lies on an edge of polygon. If none of the conditions is true, then point lies outside.



### How to handle point 'g' in the above figure?

Note that we should return true if the point lies on the line or same as one of the vertices of the given polygon. To handle this, after checking if the line from 'p' to extreme intersects, we check whether 'p' is colinear with vertices of current line of polygon. If it is colinear, then we check if the point 'p' lies on current side of polygon, if it lies, we return true, else false.

Following is C++ implementation of the above idea.

```
// A C++ program to check if a given point lies inside a given polygon
// Refer http://www.geeksforgeeks.org/check-if-two-given-line-segments-intersect/
// for explanation of functions onSegment(), orientation() and doIntersect()
#include <iostream>
using namespace std;

// Define Infinite (Using INT_MAX caused overflow problems)
```

```

#define INF 10000

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)
{
    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 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
}

// Returns true if the point p lies inside the polygon[] with n vertices
bool isInside(Point polygon[], int n, Point p)
{
    // There must be at least 3 vertices in polygon[]
    if (n < 3) return false;

    // Create a point for line segment from p to infinite
    Point extreme = {INF, p.y};

    // Count intersections of the above line with sides of polygon
    int count = 0, i = 0;
    do
    {
        int next = (i+1)%n;

        // Check if the line segment from 'p' to 'extreme' intersects
        // with the line segment from 'polygon[i]' to 'polygon[next]'
        if (doIntersect(polygon[i], polygon[next], p, extreme))
        {
            // If the point 'p' is colinear with line segment 'i-next',
            // then check if it lies on segment. If it lies, return true,
            // otherwise false
            if (orientation(polygon[i], p, polygon[next]) == 0)
                return onSegment(polygon[i], p, polygon[next]);

            count++;
        }
        i = next;
    } while (i != 0);

    // Return true if count is odd, false otherwise
    return count&1; // Same as (count%2 == 1)
}

// Driver program to test above functions
int main()
{
    Point polygon1[] = {{0, 0}, {10, 0}, {10, 10}, {0, 10}};
    int n = sizeof(polygon1)/sizeof(polygon1[0]);
    Point p = {20, 20};
    isInside(polygon1, n, p)? cout << "Yes \n": cout << "No \n";

    p = {5, 5};
}

```

```

isInside(polygon1, n, p)? cout << "Yes \n": cout << "No \n";

Point polygon2[] = {{0, 0}, {5, 5}, {5, 0}};
p = {3, 3};
n = sizeof(polygon2)/sizeof(polygon2[0]);
isInside(polygon2, n, p)? cout << "Yes \n": cout << "No \n";

p = {5, 1};
isInside(polygon2, n, p)? cout << "Yes \n": cout << "No \n";

p = {8, 1};
isInside(polygon2, n, p)? cout << "Yes \n": cout << "No \n";

Point polygon3[] = {{0, 0}, {10, 0}, {10, 10}, {0, 10}};
p = {-1, 10};
n = sizeof(polygon3)/sizeof(polygon3[0]);
isInside(polygon3, n, p)? cout << "Yes \n": cout << "No \n";

return 0;
}

```

Output:

No  
 Yes  
 Yes  
 Yes  
 No  
 No

**Time Complexity:**  $O(n)$  where  $n$  is the number of vertices in the given polygon.

**Source:**

<http://www.dcs.gla.ac.uk/~pat/52233/slides/Geometry1x1.pdf>

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  \$\text{pow}\(x, y\)\$](#)

Tags: [geometric algorithms](#), [MathematicalAlgo](#)



Writing code in comment? Please use [ideone.com](http://ideone.com) and share the link here.

39 Comments

GeeksforGeeks

Login ▾

Recommend

Share

Sort by Newest ▾



Join the discussion...



**Deepesh** · 5 months ago

Please explain this line

" If it is coliear, then we check if the point 'p' lies on current side of polygon, if it lies, we return true, else false."

I have read 3-4 times, not getting the point

^ | ▾ · Reply · Share ▸



**Ashutosh** · 6 months ago

i given vertex (1,1) ; (2,3); (5,4); (4,2) and point for testing are (2,2), (3,2); (3,3), (4,3) which are all inside but code tell yes only for first two point and then say no to other two points also i change the input order of vertex and got another result for same test points please solve this issue.

^ | ▾ · Reply · Share ▸



**Ionut Cristian** · 7 months ago

Hello! Take a look at the picture of the link:

<http://i.imgur.com/NA1ny7B.jpg>

Yellow points are points that have been returned as be inside polygon. you can see, algorithm has returned and some lines outside the polygon selected. Help me please fix this bug.

^ | ▾ · Reply · Share ▸



**Abhimanyu Dogra** · a year ago

The algorithm is still incorrect.

As couple of users below mentioned, if the "extreme" ray passes through a vertex of the polygon, 2 intersections are counted. Which makes the algorithm automatically assume the point lies outside although the point CAN lie inside.

Example :

For polygon {"1,4", "1,6", "2,7", "4,7", "5,6", "2,3"}

Point "3,6" is inside but the algorithm assumes it to be outside.

^ | ▾ · Reply · Share ▸



**Abhimanyu Dogra** → Abhimanyu Dogra · a year ago



From wikipedia : The issue is solved as follows: If the intersection point is a vertex of a tested polygon side, then the intersection counts only if the second vertex of the side lies below the ray. This is effectively equivalent to considering vertices on the ray as lying slightly above the ray.

This is how you fix it :-

replace the following piece of code in the section tagged "general case" :-

```
if (o1 != o2 && o3 != o4)
return true;
```

with :- (This code now detects intersection ONLY if the second second vertex of edge is below the extreme ray)

```
if (o1 != o2 && o3 != o4)
{
if (!o4)
```

---

see more

2 ^ | v • Reply • Share ›



**CNatka** • a year ago

The code has a flaw- If the point lies outside of the polygon but one of the vertex lies on the ray then it will return true where the expectation is to get false from the function.

For Example : Polygon {(3,6), (1,0), (5,5)} and point (0,5).

^ | v • Reply • Share ›



**anki\_2189** → CNatka • 8 months ago

Yes you are right **@CNatka** .The code has a flaw **@GeeksforGeeks**

^ | v • Reply • Share ›



**Piyush** • a year ago

what if segment joining point and extreme pass through the vertex ?. for e.g. if our polygon is Point polygon1[] = {{2,-2}, {4,0}, {2, 2}, {0, 0}}; and point is (2,0) it gives NO for that case. So condition for this case should be added.

^ | v • Reply • Share ›



**Prashanth Reddy** • a year ago

Thanks for the info... :)

^ | v • Reply • Share ›



**akshat** • a year ago

what if the coordinates are very large(order of  $10^9$ ) then the multiplication in the orientation



what if the coordinates are very large (order of  $10^9$ ) then the multiplication in the orientation method may lead to multiplication of 2 very large coordinates leading to result of a number of order of  $10^{18}$  which cannot be stored in long long int also. i tried to solve it by using the actual equation given in the slides i.e i actually calculated two slopes and subtracted them as follows  $(y_2 - y_1 / x_2 - x_1) - (y_3 - y_2 / x_3 - x_2)$  but this sol'n is still not working maybe because slopes cannot be compared like this. please sort this out

^ | v • Reply • Share ›



**Deven Bhooshan** • a year ago

There is a problem in the algorithm .

Suppose that a point to be checked lies inside the polygon and passes through only one vertex then the above algorithm will fail.

for example suppose there is a polygon having five vertices as following

```
0 0
0 4
4 2
1 1
1 0
```

Then if we check Points (1 2) (2 2) (3 2) above algorithm will not work.

4 ^ | v • Reply • Share ›



**Nishchay Kala** → Deven Bhooshan • a year ago

A bit of change is required in code, algorithm is correct though. Nice Observation :)

2 ^ | v • Reply • Share ›



**Deepesh** → Nishchay Kala • 5 months ago

plz suggest the change

^ | v • Reply • Share ›



**Deven Bhooshan** → Nishchay Kala • a year ago

yup

^ | v • Reply • Share ›



**trying** • 2 years ago

what abt the point which is inside the polygon but the ray passes through only one vertex then the algo will return outside because of even while it is inside. please check.

5 ^ | v • Reply • Share ›



**GeeksforGeeks** → trying • 2 years ago

Thanks for pointing this out. We have added this condition to algorithm and code.

^ | v • Reply • Share ›





**clotho** · 2 years ago

How about a figure 8 polygon rotated 90 degrees? A line from a point to the right crosses 2 times and its in the second loop.

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

1 ^ | v · Reply · Share ›



**badri** · 2 years ago

For the first polygon ,

Point polygon1[] = {{0, 0}, {10, 0}, {10, 10}, {0, 10}};

say point p(-1,10) , the line intersects 3 times {10, 0} to {10, 10} , {10, 10}to {0, 10} , {0, 10} to {0, 0} , by the theorem if its odd then point lies inside the polygon , need to handle this case as well..

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

2 ^ | v · Reply · Share ›



**GeeksforGeeks** → badri · 2 years ago

Thanks for pointing this out. We have made changes to handle this case. Now to handle all cases where a point is coliear with sides of polygon, we do following:

If the point lies on the side, we return true, else false. We use onSegment() for checking whether the point lies or not.

Keep it up!

^ | v · Reply · Share ›



**hh** · 2 years ago

@geeksforgeeks : Your algorithm has bug.

Consider a triangle and say point lie on one side of triangle .This point has two intersection points i.e. even so it should lie outside(acc. to your algo.) but it lie on the side i.e. inside.

Please take a look at this.Thanks.

^ | v · Reply · Share ›



**kartik** → hh · 2 years ago

I don't understand your question. How can a line from inside point to infinite can intersect with 2 sides of triangle.

See following diagram for example. The line intersects only with one line.



^ | v • Reply • Share ›



hh → kartik • 2 years ago

see diagram below:



here point lie on side ab of triangle and it intersect triangle at two points(on side ab and ac).

^ | v • Reply • Share ›



kartik → hh • 2 years ago

Thanks for providing this case. I think, we should check for onSegment rather than same point. Following is updated code. We will soon update the post with this code.

```
// Returns true if the point p lies inside the polygon[] with n vert
bool isInside(Point polygon[], int n, Point p)
{
    // There must be at least 3 vertices in polygon[]
    if (n < 3) return false;

    // Create a point for line segment from p to infinite
    Point extreme = {INF, p.y};

    // Count intersections of the above line with sides of polygon
    int count = 0, i = 0;
    do
    {
        if (doIntersect(polygon[i], polygon[(i+1)%n], p, extreme))
```

[see more](#)

^ | v • Reply • Share ›



GeeksforGeeks → kartik • 2 years ago

@hh: We have updated the code to handle the cases where point lies on a side of polygon. Keep it up!

^ | v • Reply • Share ›

**mohit** · 2 years ago

```
bool onSegment(Point p, Point q, Point r)
{
    if (q.x == min(p.x, r.x) &&
        q.y == min(p.y, r.y))
        return true;
    return false;
}
```

this is wrong . consider (1,2) , (2,4) , (4,5) are not collinear best way can be to check area of triangle = 0 of 3 points then 3 points are collinear

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

1 ^ | v · Reply · Share ›

**kartik** → mohit · 2 years ago

Take a closer look at the usage and working of onSegment(p, q, r). This is used when we have already checked if the three points are collinear. In this function, we only want to check if q lies on segment pr or not.

^ | v · Reply · Share ›

**Chaitanya** · 2 years ago

I think the best solution would be,

Draw two line from the given point 1 Horizontal & 1 Vertical.

Considering given point as origin , all 4 lines originating from the origin should have minimum 1 intersect with the polygon.

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

^ | v · Reply · Share ›

**kartik** → Chaitanya · 2 years ago

This will not work. Consider the diagram of inside example. If we put a point where "Inside example" is written, the lines passing through point will intersect, but point is not inside.

^ | v · Reply · Share ›

**Chaitanya** → kartik · 2 years ago

@Kartik

In case of Inside Example, if you draw a point outside polygon, then there wont be any case in which all the 4 lines originating from the point intersect a polygon at least a single time. In fact I didnt find any case. Please help me if u find any :S

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

^ | v • Reply • Share ›



**Chaitanya** → kartik • 2 years ago

only 3 out of 4 lines will get intersect if the point is outside in the case of "inside example".

for point to be inside, all 4 lines originating from that point should have minimum 1 intersect to polygon.

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

^ | v • Reply • Share ›



**bit\_cracker007** • 2 years ago

Input 1:

3 // no of points

1 1

2 4

2 3

2 2 // x,y coordinates of point to check for inside or outside

Input 2:

5

2 1

8 1

11 4

8 7

2 7

10 2

Output must be :

1> Inside

2> Outside

But above code produces:

1> Outside

2> Outside

^ | v • Reply • Share ›



**kartik** → bit\_cracker007 • 2 years ago

I draw a diagram for your first test case. The point (2, 2) seems to be the outside the triangle formed by (1, 1), (2, 4) and (2, 3).

^ | v • Reply • Share ›

**bit\_cracker007** → kartik · 2 years ago

That was a typo, check for the below one:

3

1 1

2 3

2 1

2 2

^ | v · Reply · Share ›

**GeeksforGeeks** → bit\_cracker007 · 2 years ago

Thanks for providing this case. This seems to have happened due to integer overflow. If we replace INT\_MAX with 200 or some other big number that doesn't cause overflow, we get the correct output. We will update the post soon.

^ | v · Reply · Share ›

**Hitesh** · 2 years ago

Here's another algorithm for the same purpose of checking whether the candidate point lies inside the given polygon or not.

Taking a small example for better understanding the algorithm used in the code:

(1) Suppose, we are given 4 polygon points in clockwise order : A, B, C, D

(2) Selected candidate point is : P

(3) Say,  $\angle ABC$  = angle made by the line segments AB and BC

(4) Consider the following 4 conditions ( 1 condition per each polygon angle ) :

(a)  $\angle ABC \geq \angle ABP \ \&\& \ \angle ABC \geq \angle CBP$

(b)  $\angle BCD \geq \angle BCP \ \&\& \ \angle BCD \geq \angle DCP$

(c)  $\angle CDA \geq \angle CDP \ \&\& \ \angle CDA \geq \angle ADP$

(d)  $\angle DAB \geq \angle DAP \ \&\& \ \angle DAB \geq \angle BAP$

(5) If all the above conditions hold good, then the candidate point P is lying inside the polygon ABCD

[see more](#)

^ | v · Reply · Share ›

**sniper** · 2 years ago

What if the ray passes through two vertex of the polygon. This corner case needs to be handled properly.

^ | v · Reply · Share ›



**GeeksforGeeks** → sniper · 2 years ago

@sniper: Please take a closer look. If the horizontal line from p to infinite passes through two vertices, then number of intersections would be 4 which is even.  
As a side note, if the point is same as one of the vertices, then we immediately return true.

^ | v · Reply · Share ›



**sreeni** · 2 years ago

You might want to update the line:

`q.y = min(r.y, r.y))`

to

`q.y = min(p.y, r.y))`

^ | v · Reply · Share ›



**GeeksforGeeks** → sreeni · 2 years ago

@sreeni: Thanks for pointing this out. We have updated the code.

^ | v · Reply · Share ›

---

 [Subscribe](#)

 [Add Disqus to your site](#)

 [Privacy](#)

**DISQUS**

Google™ Custom Search

- 
- 
- 
- - [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](#)

•

@geeksforgeeks, [Some rights reserved](#) [Contact Us!](#)

Powered by [WordPress](#) & [MooTools](#), customized by geeksforgeeks team