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

Source: https://www.geeksforgeeks.org/how-to-check-if-a-given-point-lies-inside-a-polygon/

How to check if a given point lies inside or outside a polygon? Last Updated : 23 Jul, 2025 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. Examples: Recommended Problem Please solve it on PRACTICE first, before moving on to the solution Solve Problem Approach: The idea to solve this problem is based on How to check if two given line segments intersect , and to be used as follows: Draw a horizontal line to the right of each point and extend it to infinity Count the number of times the line intersects with polygon edges. 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 the 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 collinear with vertices of the current line of polygon. If it is collinear, then we check if the point 'p' lies on current side of polygon, if it lies, we return true, else false. Following is the implementation of the above approach: C++ // C++ program for the above approach #include <bits/stdc++.h> using namespace std ; struct Point { double x , y ; }; // Checking if a point is inside a polygon bool point_in_polygon (Point point , vector < Point > polygon) { int num_vertices = polygon . size (); double x = point . x , y = point . y ; bool inside = false ; // Store the first point in the polygon and initialize // the second point Point p1 = polygon [0], p2 ; // Loop through each edge in the polygon for (int i = 1 ; i <= num_vertices ; i ++) { // Get the next point in the polygon p2 = polygon [i % num_vertices]; // Check if the point is above the minimum y // coordinate of the edge if (y > min (p1 . y , p2 . y)) { // Check if the point is below the maximum y // coordinate of the edge if (y <= max (p1 . y , p2 . y)) { // Check if the point is to the left of the // maximum x coordinate of the edge if (x <= max (p1 . x , p2 . x)) { // Calculate the x-intersection of the // line connecting the point to the edge double x_intersection = (y - p1 . y) * (p2 . x - p1 . x) / (p2 . y - p1 . y) + p1 . x ; // Check if the point is on the same // line as the edge or to the left of // the x-intersection if (p1 . x == p2 . x || x <= x_intersection) { // Flip the inside flag inside = ! inside ; } } } } // Store the current point as the first point for // the next iteration p1 = p2 ; } // Return the value of the inside flag return inside ; } // Driver code int main () { // Define a point to test Point point = { 150 , 85 }; // Define a polygon vector < Point > polygon = { { 186 , 14 }, { 186 , 44 }, { 175 , 115 }, { 175 , 85 } }; // Check if the point is inside the polygon if (point_in_polygon (point , polygon)) { cout << "Point is inside the polygon" << endl ; } else { cout << "Point is outside the polygon" << endl ; } return 0 ; } Java import java.awt.geom.Path2D ; import java.awt.geom.Point2D ; import java.util.ArrayList ; class Point { double x , y ; public Point (double x , double y) { this . x = x ; this . y = y ; } } public class PointInPolygon { // Checking if a point is inside a polygon public static boolean pointInPolygon (Point point , ArrayList < Point > polygon) { Path2D path = new Path2D . Double (); // Move to the first point in the polygon path . moveTo (polygon . get (0). x , polygon . get (0). y); // Connect the points in the polygon for (int i = 1 ; i < polygon . size (); i ++) { path . lineTo (polygon . get (i). x , polygon . get (i). y); } // Close the path path . closePath (); // Create a Point2D object for the test point Point2D testPoint = new Point2D . Double (point . x , point . y); // Check if the test point is inside the polygon return path . contains (testPoint); } // Driver code public static void main (String [] args) { // Define a point to test Point point = new Point (150 , 85); // Define a polygon ArrayList < Point > polygon = new ArrayList <> (); polygon .

add (new Point (186 , 14)); polygon . add (new Point (186 , 44)); polygon . add (new Point (175 , 115)); polygon . add (new Point (175 , 85)); // Check if the point is inside the polygon if (pointInPolygon (point , polygon)) { System . out . println ("Point is inside the polygon"); } else { System . out . println ("Point is outside the polygon"); } } } Python3 class Point : def __init__ (self , x , y): self . x = x self . y = y # Checking if a point is inside a polygon def point_in_polygon (point , polygon): num_vertices = len (polygon) x , y = point . x , point . y inside = False # Store the first point in the polygon and initialize the second point p1 = polygon [0] # Loop through each edge in the polygon for i in range (1 , num_vertices + 1): # Get the next point in the polygon p2 = polygon [i % num_vertices] # Check if the point is above the minimum y coordinate of the edge if y > min (p1 . y , p2 . y): # Check if the point is below the maximum y coordinate of the edge if y <= max (p1 . y , p2 . y): # Check if the point is to the left of the maximum x coordinate of the edge if x <= max (p1 . x , p2 . x): # Calculate the x-intersection of the line connecting the point to the edge x_intersection = (y - p1 . y) * (p2 . x - p1 . x) / (p2 . y - p1 . y) + p1 . x # Check if the point is on the same line as the edge or to the left of the x-intersection if p1 . x == p2 . x or x <= x_intersection : # Flip the inside flag inside = not inside # Store the current point as the first point for the next iteration p1 = p2 # Return the value of the inside flag return inside # Driver code if __name__ == "__main__" : # Define a point to test point = Point (150 , 85) # Define a polygon polygon = [Point (186 , 14), Point (186 , 44), Point (175 , 115), Point (175 , 85)] # Check if the point is inside the polygon if point_in_polygon (point , polygon): print ("Point is inside the polygon") else : print ("Point is outside the polygon") C# using System ; using System.Collections.Generic ; class Program { // Define a structure to represent a point struct Point { public double x , y ; } // Function to check if a point is inside a polygon static bool PointInPolygon (Point point , List < Point > polygon) { int numVertices = polygon . Count ; double x = point . x , y = point . y ; bool inside = false ; // Store the first point in the polygon and initialize the second point Point p1 = polygon [0], p2 ; // Loop through each edge in the polygon for (int i = 1 ; i <= numVertices ; i ++) { // Get the next point in the polygon p2 = polygon [i % numVertices]; // Check if the point is above the minimum y coordinate of the edge if (y > Math . Min (p1 . y , p2 . y)) { // Check if the point is below the maximum y coordinate of the edge if (y <= Math . Max (p1 . y , p2 . y)) { // Check if the point is to the left of the maximum x coordinate of the edge if (x <= Math . Max (p1 . x , p2 . x)) { // Calculate the x-intersection of the line connecting the point to the edge double xIntersection = (y - p1 . y) * (p2 . x - p1 . x) / (p2 . y - p1 . y) + p1 . x ; // Check if the point is on the same line as the edge or to the left of the x-intersection if (p1 . x == p2 . x || x <= xIntersection) { // Flip the inside flag inside = ! inside ; } } } } // Store the current point as the first point for the next iteration p1 = p2 ; } // Return the value of the inside flag return inside ; } // Driver code static void Main (string [] args) { // Define a point to test Point point = new Point { x = 150 , y = 85 }; // Define a polygon List < Point > polygon = new List < Point > { new Point { x = 186 , y = 14 }, new Point { x = 186 , y = 44 }, new Point { x = 175 , y = 115 }, new Point { x = 175 , y = 85 } }; // Check if the point is inside the polygon if (PointInPolygon (point , polygon)) { Console . WriteLine ("Point is inside the polygon"); } else { Console . WriteLine ("Point is outside the polygon"); } } } JavaScript // JavaScript program with the same function and variable names as the C++ code class Point { constructor (x , y) { this . x = x ; this . y = y ; } } function point_in_polygon (point , polygon) { const num_vertices = polygon . length ; const x = point . x ; const y = point . y ; let inside = false ; let p1 = polygon [0]; let p2 ; for (let i = 1 ; i <= num_vertices ; i ++) { p2 = polygon [i % num_vertices]; if (y > Math . min (p1 . y , p2 . y)) { if (y <= Math . max (p1 . y , p2 . y)) { if (x <= Math . max (p1 . x , p2 . x)) { const x_intersection = ((y - p1 . y) * (p2 . x - p1 . x)) / (p2 . y - p1 . y) + p1 . x ; if (p1 . x === p2 . x || x <= x_intersection) { inside = ! inside ; } } } } p1 = p2 ; } return inside ; } const point = new Point (150 , 85); const polygon = [new Point (186 , 14), new Point (186 , 44), new Point (175 , 115), new Point (175 , 85)]; if (point_in_polygon (point , polygon)) { console . log ("Point is inside the polygon"); } else { console . log ("Point is outside the polygon"); } Output Point is outside the polygon Time Complexity: O(n) where n is the number of vertices in the given polygon. Auxiliary Space: O(1), since no extra space has been taken. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above Comment Article Tags: Article Tags: Mathematical Geometric DSA Algorithms-InsertionSort TCS-coding-questions + 1 More