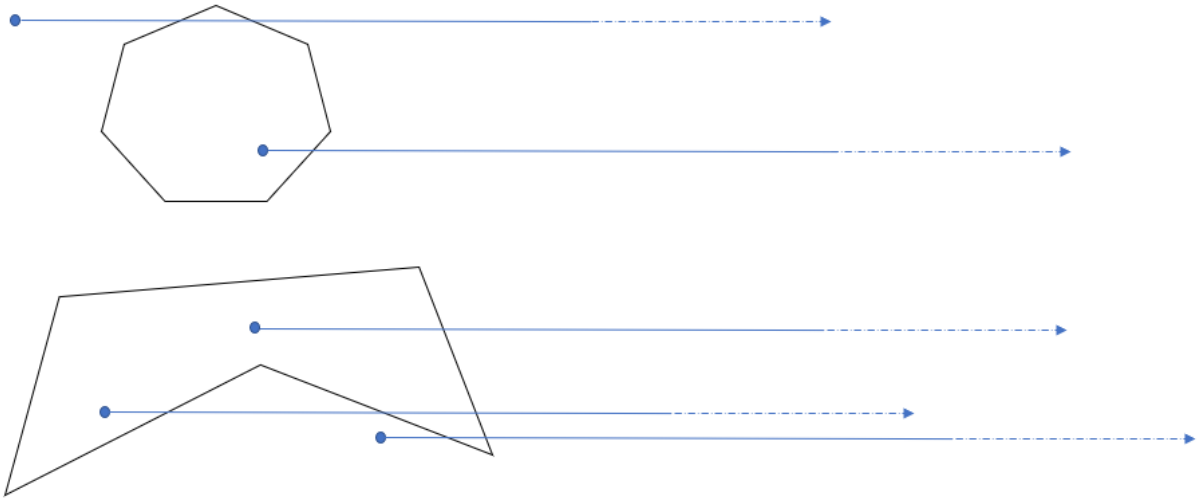
Explanation Q 1

Q-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.

The logic behind the solution is that if the point is inside the polygon, then if we draw a line joining this point with any other point outside the polygon, then it will intersect the polygon odd number of times. If the point is outside the polygon, then the line joining this point with any other point outside the polygon will intersect the polygon even number of times (including 0). The case of point being on any edge will be handled separately.

We take a point very far away from the given polygon and join this to the given point. Now we count the number of intersections it has with the edges of polygon. If the count is odd, then the point is inside, else outside.



For all points inside the curve, there are odd intersections. This logic is applicable to any polygon.

The pseudocode is-

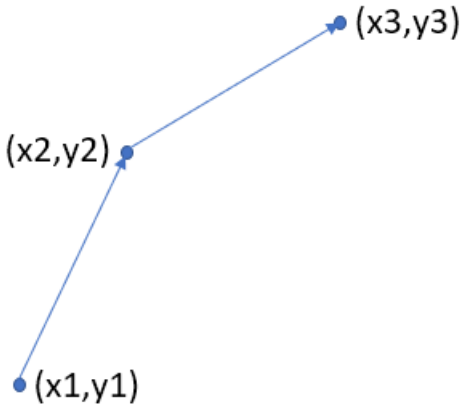
- Loop through all edges
- Count the number of times the line joining the given point to the reference point intersect and if odd, the point lies inside the polygon, else outside.

If the point lies on any edge, then both even and odd number of intersections is possible. This will be checked whenever we find that our line intersects any edge. For finding if two lines intersect, we use the concept of orientation.

Orientation of three points-

If are given three points in order, then their orientation can be in clockwise, anticlockwise sense or they can be collinear. We check the orientation using the concept of slope.

If the slope of second line is less than that of the first line, then there is a clockwise turn. If the slope increases, then there is an anticlockwise turn. Equal slope implies collinearity.



Using the above concept, for clockwise sense

$\frac{y2-y1}{x2-x1} > \frac{y3-y2}{x3-x2}$, or

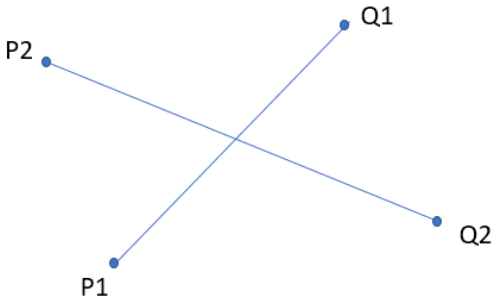
$(y2-y1) \times (x3-x2) - (y3-y2) \times (x2-x1) > 0$

This expression will be negative for anticlockwise sense and 0 for collinearity.

Checking if 2 lines intersect or not

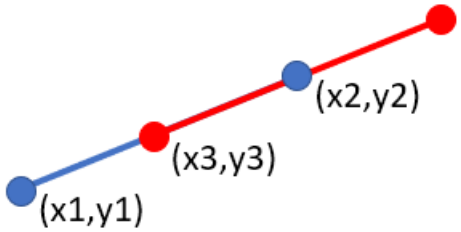
We can use the concept of orientation of 3 points to check for intersection. Two lines intersect if the orientation of the two ends of one line are opposite to each other with respect to the end points of other line.

In this figure, the orientation sense of (P1, Q1, P2) and (P1, Q1, Q2) are opposite. Similarly, the orientations of (P2, Q2, P1) and (P2, Q2, Q1) are opposite.



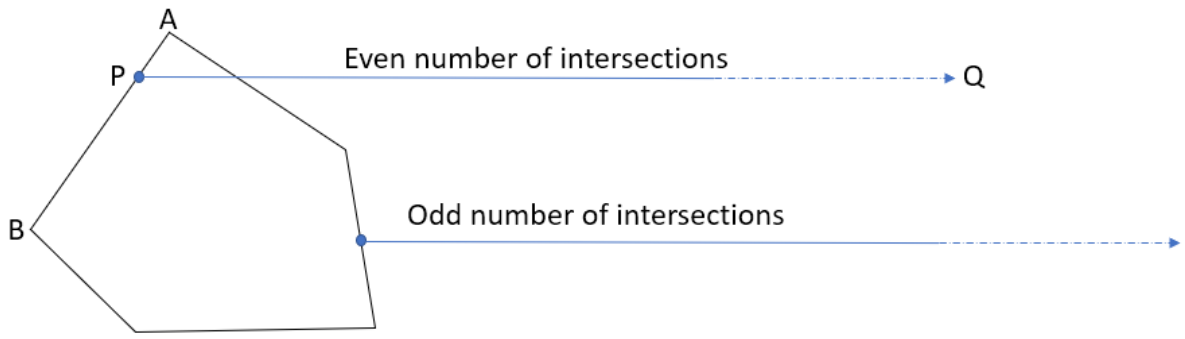
When the lines are parallel or coinciding, the orientations will be same for points of a line with respect to other line. In that case, we explicitly check if the lines are coinciding or not.

If the lines are parallel and coinciding, then x3 lies in between (x1, x2) and y3 lies in between (y1, y2).



The case of point being on any edge of polygon

If the point is on any edge of the polygon, then simply counting the number of intersections would not work. Intersection count can be both positive and negative.



Therefore, whenever we find that line PQ intersects with edge AB, we check for collinearity of the points A, P and B. If these points are collinear and P lies in between A and B, then the point is inside the polygon. Conversely, if these points are collinear but P does not lie in between A and B, then P is outside the polygon. After we have found that the points are collinear, no more edges need to be checked. We break from our algorithm after this.

We can summarize the algorithm as follows-

1- For each edge->

- Check for intersection. If intersection is found-
    - Check for collinearity. If found-
        - If the point lies in between the end points of edge, return yes.
        - Else return no.
    - Else update the count of intersecting lines

2- If the count is odd, return yes.

3- Else return no.