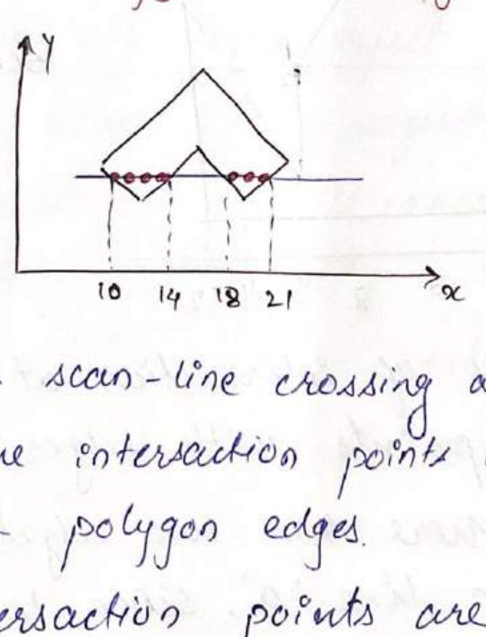
Filled - Area Primitives

- In practical we often use polygon which are filled with some color or pattern inside it.
- There are two basic approaches to area filling on raster systems.
- One way to fill area is to determine the overlap intervals for scan line that cross the area.
- Another method is to fill the area is to start from a given interior position and paint out wards from this point untill we encounter boundary.

Scan - line Polygon Fill Algorithm.



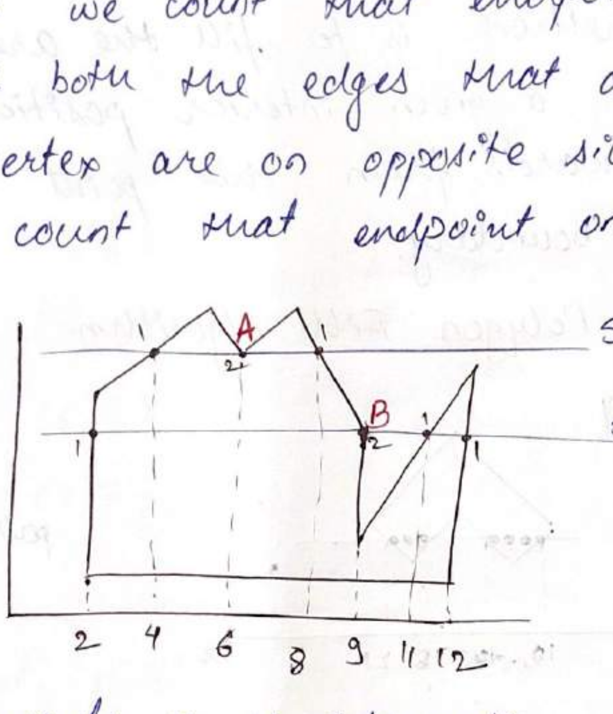
pair (10, 14)
(18, 21)

- For each scan-line crossing a polygon, the algo. locates the intersaction points are of scan line with the polygon edges.
- This intersaction points are stored from left to right.
- The pair of intersaction are marked and fill color within all the pixels inside the pair

- Some scan line intersection at polygon vertices require special handling.
- A scan line passing through a vertex intersects two polygon edges at that position, adding two points to the list of intersection for the scan line.

  - If both the edges that are connected to the vertex are on same side of scan line then we count that endpoint twice.

  - If both the edges that are connected to the vertex are on opposite side of scan line we count that endpoint once.



12- represents no. of intersaction at that particular points with edges.

- At point A there are two edges intersecting with the scan line 'a'. since both the edges are on same side of scan line it upside, will count the endpoint twice. Hence the pair that we get for scan line 'a' is {(4,6) & (6,8)}

- At point B there are two edges intersecting with the scan line 'b'. Since both the edges are on opposite side of scan line, we will count the point once.

Hence the pair that we get for scan line 'b' are (2, 9) & (11, 12)

Inside-Outside Test.

- Area filling algorithm and other graphics processes often need to identify interior region of objects.
- It is not always clear which region of my plan would be interior and which region would be exterior to the object.
- This is because in algorithm we can give the vertices of the fill area in any order which does not specify which region is interior and which is exterior.
- There are two methods to find out if the given region is interior or exterior

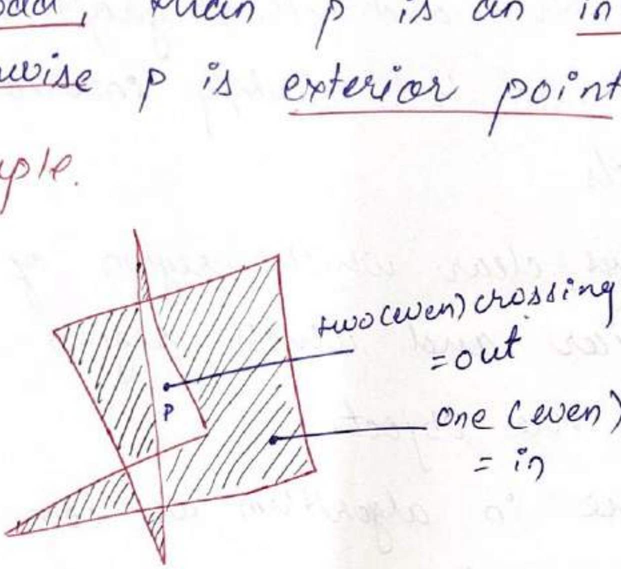
1) odd- even rule / odd parity rule

2) Non zero winding number.

Odd-even rule:

- It is also called the odd parity rule or even odd rule
- By conceptually drawing a line from any position p to a distant point outside the co-ordinate extents of the object and counting the number of edges crossing by this line is odd, than p is an interior point, otherwise p is exterior point.

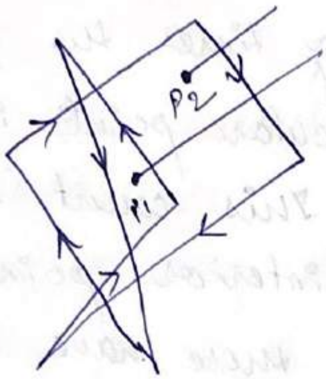
example.



- To obtain accurate edge count we must sure that line selected is not passing from any vertex.

Non zero winding number rule

- This method counts the no. of times the polygon edges wind around a particular point in the counterclockwise direction. This count is called the winding no. and the interior points of a 2D object are defined to be those have a nonzero value for the winding number.
- We apply this rule by initializing winding no. with 0 and then draw a line for any point P to distant point beyond the co-ordinate extents of the object.
- The line we choose must not pass through vertices.
- Then we move along that line we find no. of intersecting edges. and we add 1 to winding no. everytime we intersect a polygon edge that crosses the line from counter-clockwise direction, and we subtract 1 everytime we intersect an edge that crosses line from clockwise direction.
- The final value of winding no. is nonzero then the point is interior and if winding no. is zero the point is exterior.



winding no. of P_1

$= +1 -1 = 0$

Hence P_1 is exterior

winding no. of P_2

$= -1$

Hence P_2 is interior