FRANKLIN'S LECTURES

COMPUTER GRAPHICS & IMAGE PROCESSING

Module 2 Part 3

CST304

POLYGON FILLING ALGORITHMS

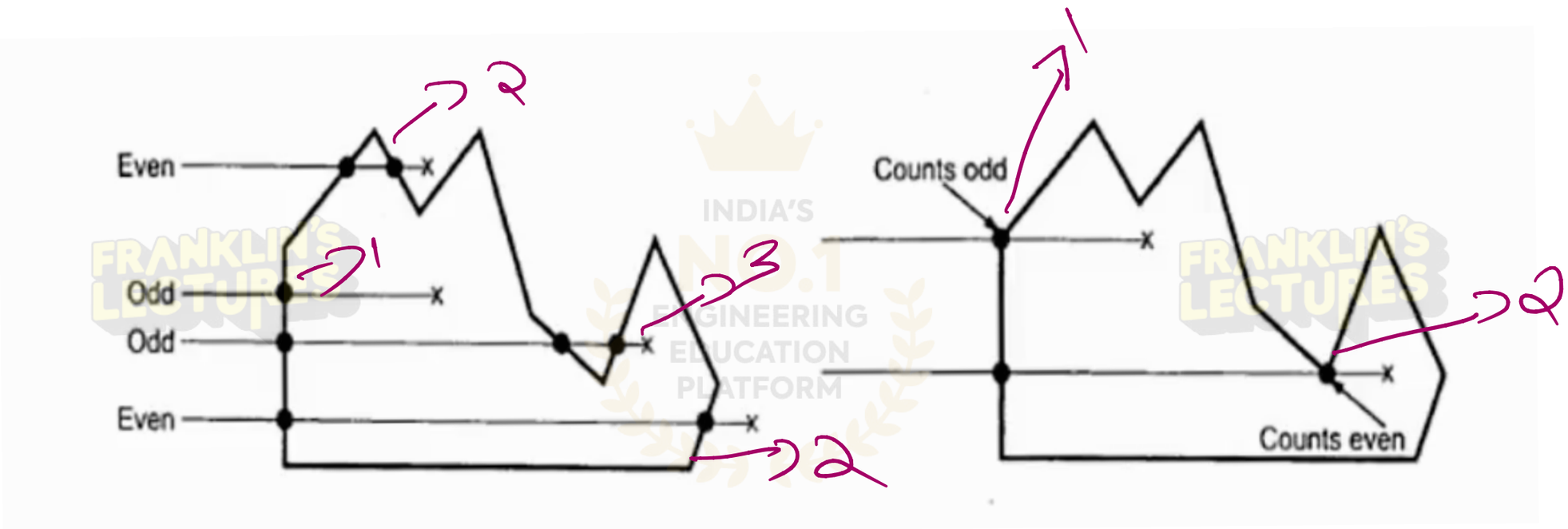
- In geometry, a polygon can be defined as a flat or plane, two-dimensional closed shape bounded with straight sides. It does not have curved sides.

Filled Area Primitives-

Area filling algorithms often need to identify interior regions of objects.

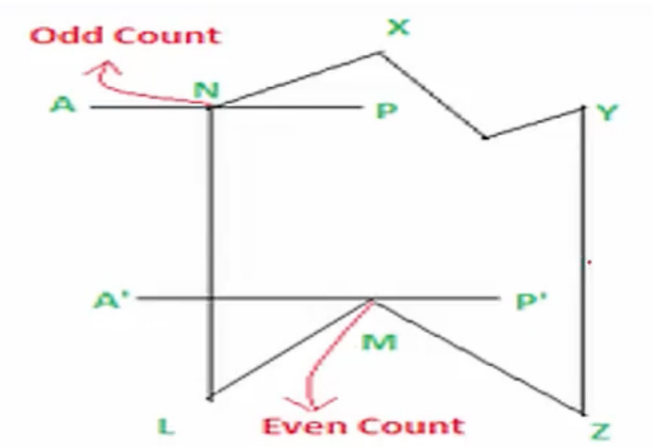
Odd Even Rule (Odd Parity Rule)

1. Construct a line segment from point to be examined to point outside of a polygon.
2. Count the number of intersections of line segment with polygon boundaries.
3. If Odd number of intersection, then Point lies inside of Polygon.
4. Else, Point lies outside of polygon.


Even
Odd
Odd
Even
Counts odd
Counts even

SCAN LINE CUT A VERTEX IS A SPECIAL CASE;

- Consider edges intersect at a vertex, if their other endpoints are on the same side of the scan line count number of intersections as 2(or even).
- Consider edges intersect at a vertex, if their other endpoints are on the opposite sides of the scan line count number of intersections as 1(or odd)-

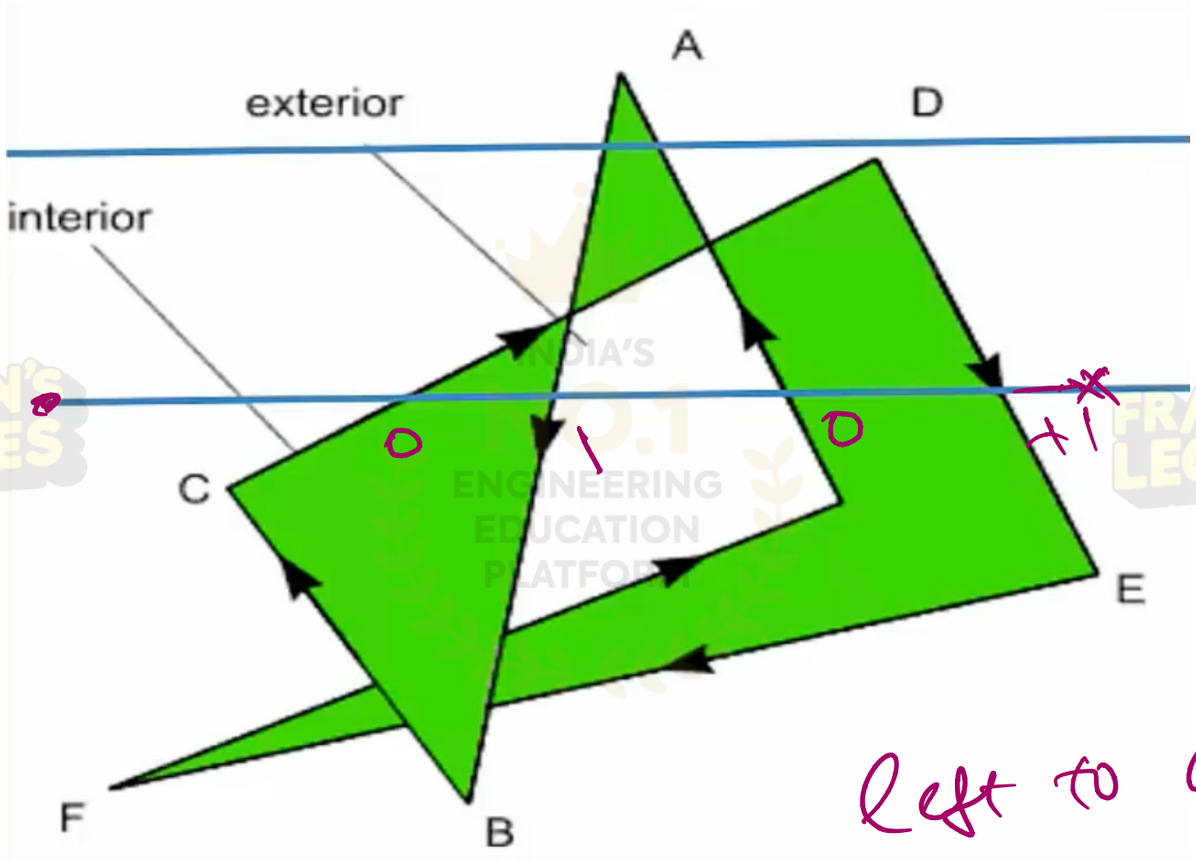


NON ZERO WINDING NUMBER RULE

FRANKLIN'S LECTURES

- Initial value of winding number=0.
- Then draw a line from point (p-point to test) to outside the polygon which does not pass through any vertex.
- Add 1 to the winding number every time we intersect a polygon edge that crosses the line from right to left and subtract 1 every time we intersect an edge that crosses from left to right.
- The interior points are that have a non zero value for the winding number.
- Exterior points are those whose value of the winding number is zero.

Count non-zero = Interior
Count zero = Exterior



Exterior Point

Left to Right = -1

Right to left = +1

BOUNDARY FILL ALGORITHM

- In Boundary fill Algorithm the basic concept is filling the color in closed area by starting at a point inside a region and paint the interior outward towards the boundary.
- One requirement for Boundary Fill algorithm is that the boundary has to have a single color.

WORKING

FRANKLIN'S LECTURES

- In Boundary fill algorithm, you start from a point inside the region and fill the color interior outward towards the boundary pixel by pixel.
- So the process is that you check the default color of the pixel before filling, if the color is not boundary color, then you fill it with the fill color, and move to the next pixel and check for the same criteria till you encounter the boundary colored pixel or the boundary.

METHODS OF IMPLEMENTATION

- There are two methods in which the Boundary Fill Algorithm can be implemented:
  1. 4-connected pixel
  2. 8-connected pixel

4-CONNECTED PIXEL

- In 4 connected pixel method you check 4 pixels adjacent to the current pixel, namely towards the left, right, top & bottom.
- So you fill the area with 4-connected pixel method by following these steps

Step 1:
First initialize the 4 values namely x, y, Fill-color & Default-color.
Where x and y are co-ordinate positions of the initial interior pixel we start the boundary fill algorithm & Fill-color is the color we want to fill and Default-color is the default color of the interior pixel

Step 2:
Define the value of the boundary pixel color or Boundary color.

Step 3:
Now check if the current pixel is of Default-color and if Yes then Repeat Step 4 and step 5 till the boundary pixels are reached.

Step 4:

Change the default color with the fill color at the current pixel.

Step 5:

Repeat step 3 and step 4 for the neighboring 4 pixels.

Step 6:

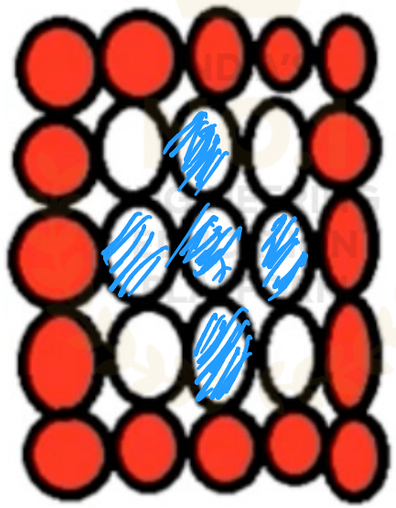
Exit.

4-CONNECTED

```
void boundaryFill4  (int x, int y,  int fill, int boundary, int default)
{
  int current;
  current =  getpixel  (x, y);
    if ((current ! =  boundary) &&  (current = default))
        {
         setPixel  (x, y, fill);
         boundaryFill4 (x+1, y, fill, default);
         boundaryFill4  (x-1, y, fill, default);
         boundaryFill4  (x, y+1, fill, default);
         boundaryFill4  (x, y-1, fill, default);
        }
}
```

→ Right
→ Left
→ Top
→ Bottom

- There is a problem with this technique that 4-connected pixels technique cannot fill all the pixels.



- 8-connected pixel
- To solve the problems that can occur in 4-connected pixel method, a new method was introduced called 8-connected pixel.
- SO in 8-connected pixel method we instead of filling just 4 adjacent pixels we fill also the adjacent diagonal pixels positions, such as (x + 1, y + 1) .

8-CONNECTED PIXEL

- To solve the problems that can occur in 4-connected pixel method, a new method was introduced called 8-connected pixel.
- So in 8-connected pixel method we instead of filling just 4 adjacent pixels we fill also the adjacent diagonal pixels positions, such as (x + 1, y + 1) .

8-CONNECTED

FRANKLIN'S LECTURES

void boundaryFill8 (int x , int y , int fill, int boundary, int default)

```
{
 int current;
  current =  getpixel  (x, y);
    if ((current ! =   boundary) &&  (current = default))
        {
         setPixel  (x, y, fill);
         boundaryFill8 (x+1, y, fill, default);
         boundaryFill8 (x-1, y, fill, default);
         boundaryFill8 (x, y+1, fill, default);
         boundaryFill8 (x, y-1, fill, default);
         boundaryFill8 (x+1, y+1, fill, default);
         boundaryFill8 (x+1, y-1, fill, default);
         boundaryFill8 (x-1, y+1, fill, default);
         boundaryFill8 (x-1, y-1, fill, default);

        }
}
```

4-connected

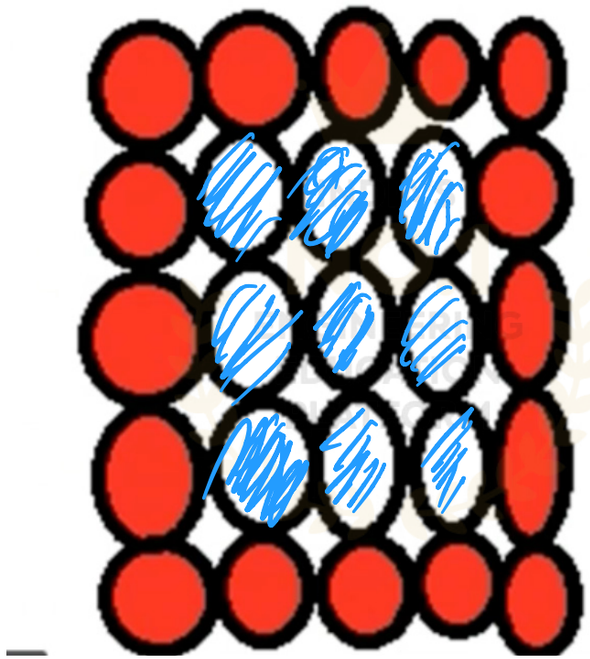
top-right

bottom left

top left

left bottom

8-CONNECTED PIXEL FILLING



FLOOD FILL ALGORITHM

- Flood fill algorithm is useful in cases where there no single color boundary for the polygon, i.e. the boundary has multiple colors.
- In flood fill algorithm instead of filling color till you encounter a specific boundary color you just fill the pixels with default color.
- It is used in the "bucket" fill tool of paint programs to fill connected, similarly-colored areas with a different color

ALGORITHM

Flood-fill (node, target-color, replacement-color):

1. If target-color is equal to replacement-color, return.
2. If the color of node is not equal to target-color, return.
3. Set the color of node to replacement-color.
4. Perform Flood-fill (one step to the south of node, target-color, replacement-color).

Perform Flood-fill (one step to the north of node, target-color, replacement-color).

Perform Flood-fill (one step to the west of node, target-color, replacement-color).

Perform Flood-fill (one step to the east of node, target-color, replacement-color).

5. Return.

METHODS OF IMPLEMENTATION

- There are two methods in which can be implemented:
  1. 4-connected pixel
  2. 8-connected pixel

4-CONNECTED PIXEL

- In 4 connected pixel method you check 4 pixels adjacent to the current pixel, namely towards the North(top), South(bottom), west(left), East(right).
- So you fill the area with 4-connected pixel method by following these steps

Step 1:
First initialize the 4 values namely x, y, Fill-color & Default-color.
Where x and y are co-ordinate positions of the initial interior pixel we start the flood fill algorithm & Fill-color is the color we want to fill and Default-color is the default color of the interior pixel

Step 2:
If the color of node is not equal to default-color, return.

Step 3:
Set the color of node to replacement-color.

Step 4:
Set the color of node to the south of node to replacement-color.

Step 5:
Set the color of node to the north of node to replacement-color.

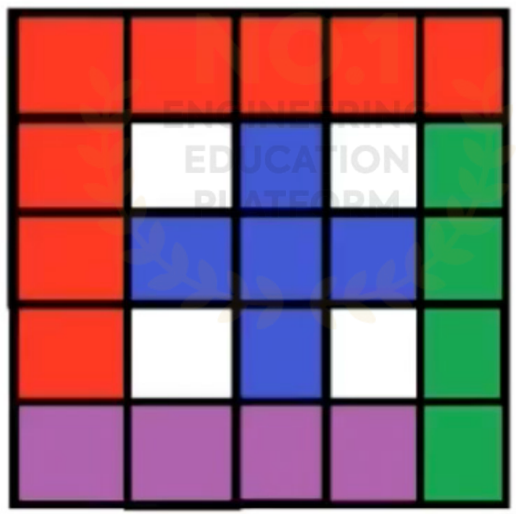
Step 6:
Set the color of node to the east of node to replacement-color.

Step 7:
Set the color of node to the west of node to replacement-color.

Step 8:
Exit.

```
Procedure floodfill ( x,  y , fill_ color, old_color: integer)
If (getpixel ( x , y ) = old_color)
{
    setpixel (x, y, fill_color);
    fill (x+1, y, fill_color, old_color);
        fill (x-1, y, fill_color, old_color);
    fill (x,y+1, fill_color, old_color);
    fill (x, y-1, fill_color, old_color);
    }
}
```

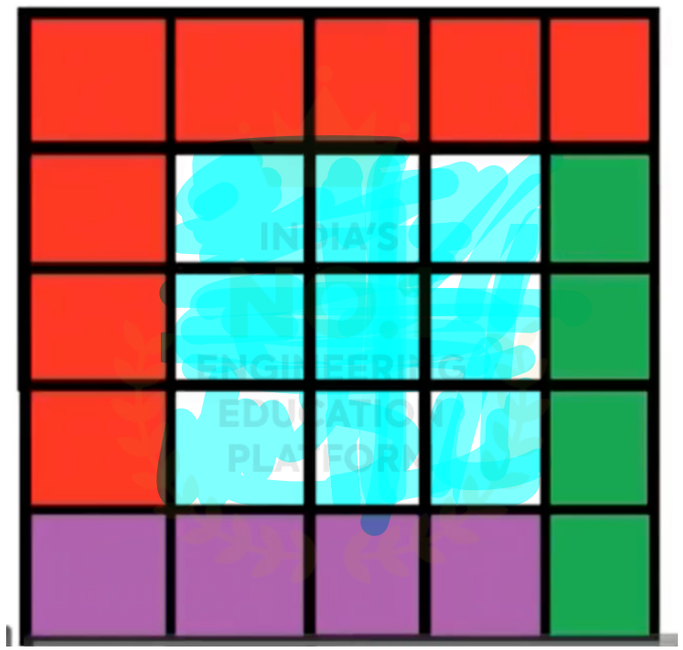
There is a problem with this technique that 4-connected pixels technique cannot fill all the pixels



8-CONNECTED PIXEL

- To solve the problems that can occur in 4-connected pixel method, a new method was introduced called 8-connected pixel.
- SO in 8-connected pixel method we instead of filling just 4 adjacent pixels we fill also the adjacent diagonal pixels positions, such as (x + 1, y + 1) .

8-CONNECTED PIXEL FILLING



SCAN LINE POLYGON FILL ALGORITHM

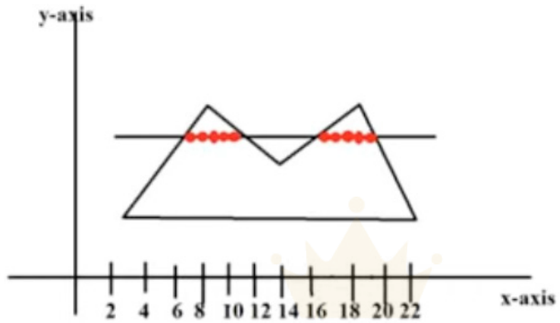
- Scan line polygon filling algorithm is used for solid color filling in polygons.

STEPS TO PERFORM:

For Scan line polygon filling there are three steps to perform in the following order:

1. Find the intersections of the scan line with all edges of the polygon.
2. Sort the intersections by increasing x-coordinate i.e. from left to right.
3. Make pairs of the intersections and fill in color within all the pixels inside the pair.

EXAMPLE

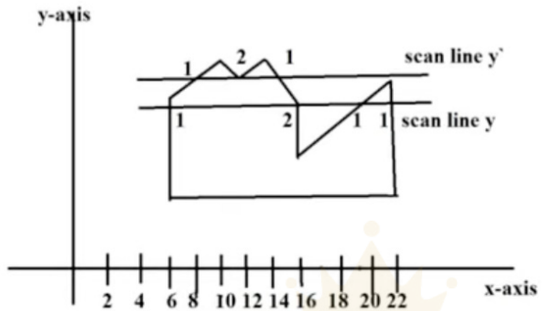


- So in our example if we follow the steps above, We will make a list of all the intersections. 6,12,16,20.
- Now lets sort our intersections by increasing x-co-ordinate.
- SO the order will be 6,12,16,20.
- Now Final step, make pair of intersections, SO we get, 2 pairs: (6,12) & (16,20).
- Now fill in all pixels with color inside the pair

Special Cases:

- Some scan-line intersections 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 intersections for the scan line.

EXAMPLE



(8,12,12,14)

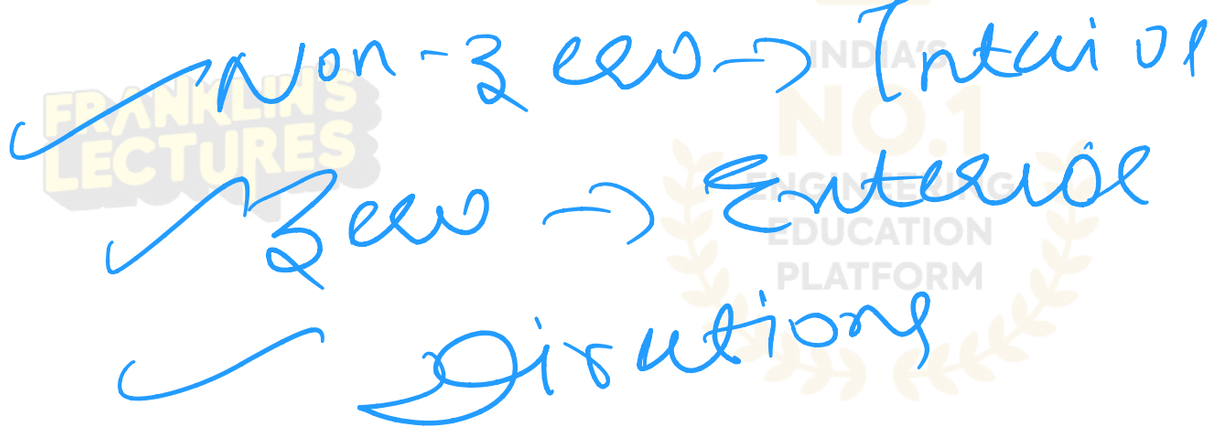
y' = (8,12) (12,14)

- In this example Scan line y and Scan line y′ both pass through an vertex or an edge endpoint.
- Now in case of scan line y′ the scan line is intersecting 4 edges i.e. even number of edges and Also passing through a vertex/ endpoint.
- ALSO both the edges that are connected to the vertex are on SAME SIDE, of the scan line. SO we need to count the vertex/ endpoint TWICE so that we can make pairs of intersection points (8,12) & (12,14).

y = (6,16) (19,22)

- Now in case of scan line y, the scan line is intersecting with 5 different edges i.e. ODD NUMBER and also passing through a vertex/endpoint. SO in this case we need to do some extra processing, i.e. we need to check if the 2 edges at the endpoint through which the scan line is passing are they on opposite sides or on same sides?
- In case of scan line y' they are on same sides and in case of scan line y both edges at the vertex are on opposite sides. SO Now we COUNT THE VERTEX AS A SINGLE INTERSECTION POINT.
- Now we just need to sort the intersection points and make pairs of them and fill all pixels which

Q. Explain the non-zero winding number rule to identify the interior regions of a polygon.
(3Marks ,JUNE 2022)



Q. Explain the Boundary Fill Algorithm, its working principle, and the step-by-step process. Compare the 4-connected and 8-connected approaches, and explain how the 8-connected approach improves over the 4-connected approach.

(8 Marks,April 2025)

- 4 connected
- 8 connected

Q. Compare the Scan Line Algorithm and the Flood Fill Algorithm used for polygon filling in computer graphics

(7Marks, April 2025)

Scan line	Flood Fill
· line by line horizontally	· Fill area from a starting point
· Does not depends on color	· Depends on color
· Pair	· Replace old color with new

Q. Write down the 4- neighbour Flood-filling algorithm

(3 Marks,JUNE 2023)

- Algorithm
- Steps

FRANKLIN'S LECTURES

THANK YOU