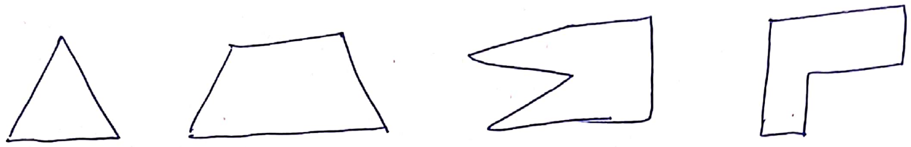
By Dr. Tanu Singh

Scan line Polygon :-

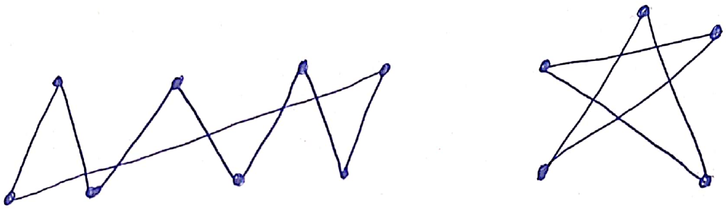
Polygon
↓
properties → It consist of line segments.
→ It must be closed circuit.

Types of Polygon :

Simple Polygon :



Complex Polygon :



⇓

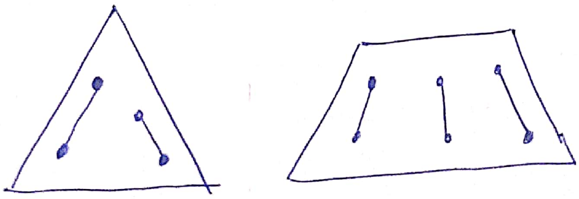
to create the complex.

⇓

Intersection crossover is must

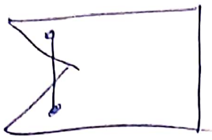
* Convex Polygon : If the two interior points of the line lies inside the polygon then it is convex

Eg :



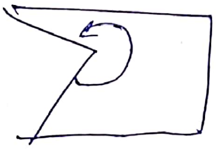
* Concave Polygon : If the two interior points of the line lies outside the polygon then it is called concave polygon

Eg :



if the angle of polygon is less than 180° then it is called as convex polygon

Else
it is called concave
greater than 180°



* Inside Outside Test Polygon :-

↳ It is used to identify the pixel / pts. is within the polygon or outside the polygon.

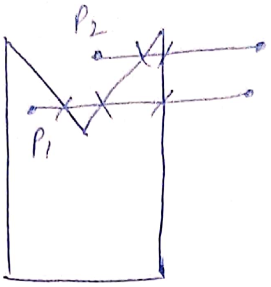
Algo of Ray Casting :-

① Draw the horizontal line from the point

② Count the no. of times the line intersect with the edge.

③ If the number of count is number == odd then it is inside
else number == even then it is outside

Eg



Point	Intersection	Lied at
P_1	3 (odd)	Inside
P_2	2 (even)	outside

∴ P_1 lies inside the polygon & P_2 lies outside the polygon

Special Case :- If the line pass / cross at the vertex 'or' crosssection point of the two lines.

We use :

Winding Number Algorithm :

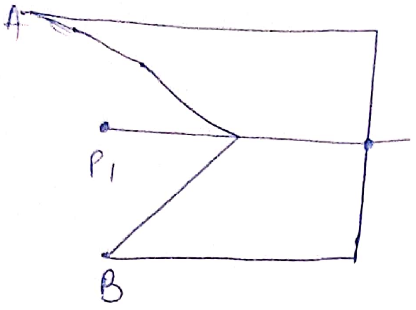
① Check the respective line segment of that point

② Check the other end of the line segment

③ check if both point is same side, consider even or different side of the line segment, consider odd.

④ Count the no. of intersection = ?.

Q1



Respective Line ⇒ not same side

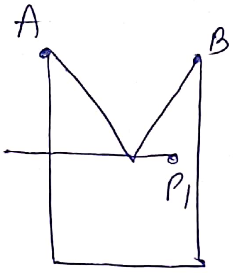
= odd

Intersection = 1

Then ⇒ 1 + odd = even.

∴ P1 lies outside the polygon

Q2



Respective Line ⇒ Same side

⇒ even

Intersection = 1

Then, 1 + even = odd

∴ P1 lies inside the polygon

Boundary fill Algorithm :-

↳ color fill algorithm

↳ Boundary color is different

↳ within the polygon color is different.

Algo :-

① Let 'P' take the point inside the polygon

② Start with co-ordinate (x, y)

③ inside color is 'F' ← fill color

④ boundary color is 'b'

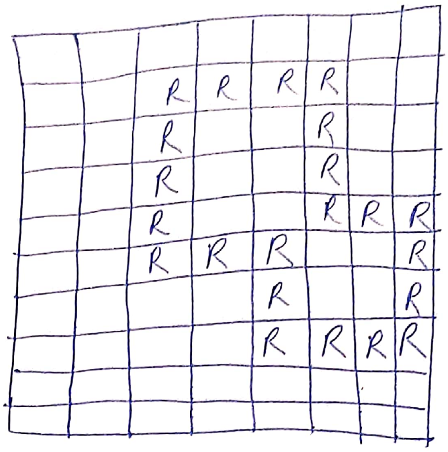
Points to remember before going to algo :-

	(x, y+1)	
(x-1, y)	(x, y)	(x+1, y)
	(x, y-1)	

4 connected

(x-1, y+1)	(x, y+1)	(x+1, y+1)
(x-1, y)	(x, y)	(x+1, y)
(x-1, y-1)	(x, y-1)	(x+1, y-1)

8 connected



```
boundary (x, y, F, b)
{
    if (getpixel (x, y) != b
              &&
         getpixel (x, y) != F)
    {
        putpixel (x, y, F)

        boundary (x+1, y, F, b)
        boundary (x, y+1, F, b)
        boundary (x-1, y, F, b)
        boundary (x, y-1, F, b)

        boundary (x-1, y-1, F, b)
        boundary (x-1, y+1, F, b)
        boundary (x+1, y+1, F, b)
        boundary (x+1, y-1, F, b)
    }
}
```

4 connected

8- connected Pixel algorithm

(4)

Dr. Tanu Singh

Flood Fill Algorithm :-

New color

old color

```
Flood (x, y, N, O)
{
    if ( getpixel (x, y) = = O)
    {
        Flood (x+1, y, N, O)
        Flood (x, y+1, N, O)
        Flood (x-1, y, N, O)
        Flood (x, y-1, N, O)

        Flood (x-1, y-1, N, O)
        Flood (x-1, y+1, N, O)
        Flood (x+1, y-1, N, O)
        Flood (x-1, y+1, N, O)
```

4-connected

8-connected pixel Algorithm

		R	R	R	R		
		R			G		
		R			G	G	R
		R	G	G			R
				G			R
				R	R	R	R

N ⇒ New color

O ⇒ Old color