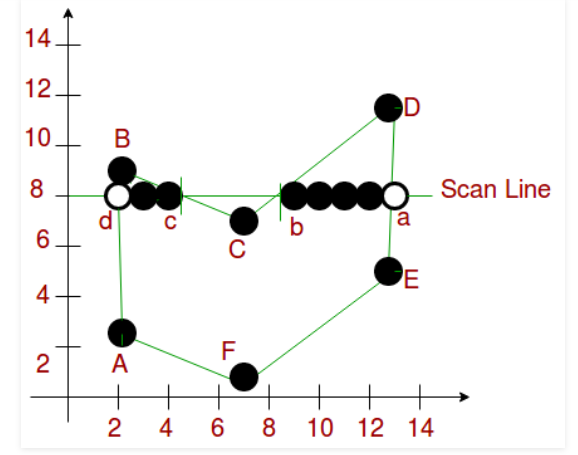
General scan-line polygon fill Algorithm

Scanline Polygon filling Algorithm

Scanline filling is basically filling up of polygons using horizontal lines or scanlines. The purpose of the Scanline Polygon filling Algorithm is to fill (color) the interior pixels of a polygon given only the vertices of the figure. To understand Scanline, think of the image being drawn by a single pen starting from bottom left, continuing to the right, plotting only points where there is a point present in the image, and when the line is complete, start from the next line and continue.  
This algorithm works by intersecting scanline with polygon edges and fills the polygon between pairs of intersections.



1. If both lines intersecting at the vertex are on the same side of the scanline, consider it as two points.
2. If lines intersecting at the vertex are at opposite sides of the scanline, consider it as only one point.

**Components of Polygon fill:**

Edge Buckets: It contains an edge’s information. The entries of edge bucket vary according to data structure you have used. In the example we are taking below, there are three edge buckets namely: ymax, xofymin,  
slope-inverse.

1. Edge Table: It consists of several edge lists -> holds all of the edges that compose the figure. When creating edges, the vertices of the edge need to be ordered from left to right and thee edges are maintained in increasing yMin order. Filling is complete once all of the edges are removed from the ET
2. Active List: IT maintains the current edges being used to fill in the polygon. Edges are pushed into the AL from the Edge Table when an edge’s ymin is equal to the current scan line being processed.  
   The Active List will be re-sorted after every pass.

Scan line polygon filling1

**Algorithm:**

1. We will process the polygon edge after edge, and store in the edge Table.

2. Storing is done by storing the edge in the same scanline edge tuple as

the lowermost point's y-coordinate value of the edge.

3. After addition of any edge in an edge tuple, the tuple is

sorted using insertion sort, according to its xofymin value.

4. After the whole polygon is added to the edge table,

the figure is now filled.

5. Filling is started from the first scanline at the bottom,

and continued till the top.

6. Now the active edge table is taken and the following things

are repeated for each scanline:

i. Copy all edge buckets of the designated scanline

to the active edge tuple

ii. Perform an insertion sort according

to the xofymin values

iii. Remove all edge buckets whose ymax is equal

or greater than the scanline

iv. Fill up pairs of edges in active tuple, if any vertex is got,

follow these instructions:

o If both lines intersecting at the vertex are on

the same side of the scanline, consider it as two points.

o If lines intersecting at the vertex are at

opposite sides of the scanline, consider it as only one point.

v. Update the xofymin by adding slope-inverse for each bucket.

Polygon

M=Yk+1 - Yk/ Xk+1 - Xk

YK+1 - Yk=1 bcoz Yk+1=Yk +1

XK+1 = Xk + 1/m, m=dy/dx, (y2-y1)/(x2-x1)

XK + 1 = Xk + dx/dy