Bresenham's Line Algorithm

An accurate and efficient raster line-generating algorithm, developed by Bresenham, scan converts lines using only incremental integer calculations that can be adapted to display circles and other curves. Sampling at unit ***x*** intervals, we need to decide which of two possible pixel positions is closer to the line path at each sample step.

For example, starting from the left endpoint (10,11) **,** we need to determine at the next sample position whether to plot the pixel at position (11, 11) or the one at (11, 12). Similarly, a negative slope-line path starting from the left endpoint at pixel position (50, 50). In this one, do we select the next pixel position as (51,50) or as (51,49)? These questions are answered with Bresenham's line algorithm by testing the sign of an integer parameter, whose value is proportional (يتناسب) to the difference between the separations of the two pixel positions from the actual line path.

Example 1:

To illustrate the algorithm, we digitize the line with endpoints (20, 10) and (30, 18). This line has a slope of 0.8, with

$$\Delta x = 10, \qquad \Delta y = 8$$

The initial decision parameter has the value

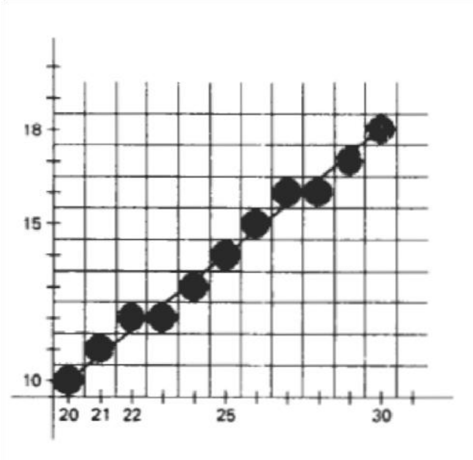
$$p_0 = 2\Delta y - \Delta x$$
$$= 6$$

and the increments for calculating successive decision parameters are

$$2\Delta y = 16, \qquad 2\Delta y - 2\Delta x = -4$$

We plot the initial point $(x_0, y_0) = (20, 10)$, and determine successive pixel positions along the line path from the decision parameter as

k	p_k	(x_{k+1}, y_{k+1})	k	p_k	(x_{k+1}, y_{k+1})
0	6	(21, 11)	5	6	(26, 15)
1	2	(22, 12)	6	2	(27, 16)
2	−2	(23, 12)	7	−2	(28, 16)
3	14	(24, 13)	8	14	(29, 17)
4	10	(25, 14)	9	10	(30, 18)



Bresenham's algorithm is generalized to lines with arbitrary slope by considering the symmetry between the various octants and quadrants of the *xy* plane. For a line with positive slope greater than 1, we interchange the roles of the x and y directions. That is, we step along they direction in unit steps and calculate successive x values nearest the line path. Also, we could revise the program to plot pixels starting from either endpoint. If the initial position for a line with positive slope is the right endpoint, both x and y decrease as we step from right to left. For negative slopes, the procedures are similar, except that now one coordinate decreases as the other increases.

Table 3.7 can be used to determine the octant of the slope. Given a line segment from (x1, y1) to (x2, y2), first reorder the points, if necessary, such that x1 ≤ x2, then use the table. The top row of the table reads: If Δy ≥ 0 and Δx ≥ Δy, then the slope is positive and is less than or equal 1. The octant is either 1 or, if the points had to be swapped, it is 5.

ΔY	ΔX?ΔY	slope	octant
≥ 0	≥	Pos ≤ 1	1 (5)
≥ 0	<	Pos >1	2 (6)
< 0	≤	Neg ≥ -1	7 (3)
< 0	>	Neg < -1	8 (4)

General Bresenham's algorithm for all Octants

Inputs: Start point (X1 , Y1) , End point (X2 , Y2)

Begin

X=X1

Y=Y1
ΔX =Abs (X2-X1)
ΔY =Abs(Y2-Y1)
S1=Sign (X2-X1)
S2=Sign (Y2-Y1)
If ΔY > ΔX Then
    T= ΔX , ΔX = ΔY , ΔY =T , Interchange=1
Else
    Interchange =0
End If
E= 2 ΔY - ΔX
A= 2 ΔY
B= 2ΔY - 2ΔX
Plot (X,Y)
For i=1 to ΔX If
    (E < 0)
        If Interchange=1 Then Y= Y + S2
        Else X=X+S1
        E= E +A
    Else Y=Y + S2
        X=X + S1
        E = E + B
    End If
    SetPixel (X,Y)
End for
End

Note : Sign function returns : -1 if its argument is < 0
: 0 if its arguments is = 0
: +1 if its arguments is > 0

Ex: Sign(-10) = -1 Sign (5) = 1

Example 2: Draw the line from (0,0) to (-8,-4) using General Bresenham algorithm

Solution : X=0 , Y=0 , ΔX =8 , ΔY=4 , A=2ΔY= 8 , B=2ΔY-2ΔX=-8 , S1=-1 , S2=-1

Because ΔX > ΔY then Interchange=0 , E=0

Iteration	E	X	Y	Plot

	0	0	0	(0,0)
1	-8	-1	-1	(-1,-1)
2	0	-2	-1	(-2,-1)
3	-8	-3	-2	(-3,-2)
4	0	-4	-2	(-4,-2)
5	-8	-5	-3	(-5,-3)
6	0	-6	-3	(-6,-3)
7	-8	-7	-4	(-7,-4)
8	0	-8	-4	(-8,-4)