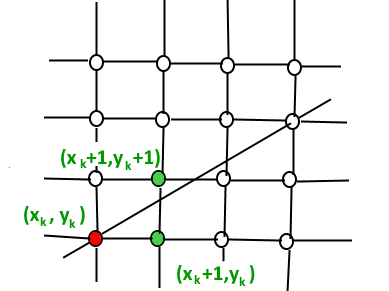
**Bresenham’s line drawing algorithm**

* The simple DDA has the disadvantages of using two operations that are expensive in computational time: floating point addition and the round function.
* The Bresenham’s algorithm is another incremental scan conversion algorithm.
* The big advantage of this algorithm is that it uses only integer calculations.
* The main Idea of the Bresenham’s line drawing algorithm ( for slope greater than equal to 1): Move across the x-axis in unit intervals and at each step choose between two different y coordinates.
* The Bresenham’s line algorithm has the following advantages:
  + An fast incremental algorithm
  + Uses only integer calculations
* The above algorithm works, but it is slow. The idea of Bresenham’s algorithm is to avoid floating point multiplication and addition to compute mx + c, and then compute the round value of (mx + c) in every step. In Bresenham’s algorithm, we move across the x-axis in unit intervals.
* We always increase x by 1, and we choose about next y, whether we need to go to y+1 or remain on y. In other words, from any position (Xk, Yk) we need to choose between (Xk + 1, Yk) and (Xk + 1, Yk + 1).
* 
* We would like to pick the y value (among Yk + 1 and Yk) corresponding to a point that is closer to the original line.
* We need a decision parameter to decide whether to pick Yk + 1 or Yk as the next point. The idea is to keep track of slope error from the previous increment to y. If the slope error becomes greater than 0.5, we know that the line has moved upwards one pixel and that we must increment our y coordinate and readjust the error to represent the distance from the top of the new pixel – which is done by subtracting one from the error.

***Algorithm***

1.Input two points (x1 ,y1 ) and (x2 ,y2 )

2. Compute dx = | x2 - x1 | & dy= |y2 - y1 |

3. If (x2 > x1 ) lx=1 else lx= - 1

4. If (y2 > y1 ) ly=1 else ly= - 1

5. Plot first point (x1 ,y1 )

6. If (Δx > Δy) (i.e. when |m| <1)

{

1. calculate p0 = 2Δy – Δx 21
2. Starting at k =0 to Δx times , repeat

If pk <0

xk+1 = xk+lx , yk+1= yk

Pk+1 = pk + 2Δy

Else

xk+1 = xk+lx ,

yk+1= yk +ly

Pk+1 = pk + 2Δy - 2Δx

}

7. Else /\* i.e. when |m|>1 \*/

{ i) Calculate p0 = 2Δx – Δy

1. Starting at k =0 to Δy times , repeat

If pk<0

xk+1 = xk

yk+1= yk+ly

Pk+1 = pk + 2Δx

else

xk+1 = xk+lx

yk+1= yk +ly

Pk+1 = pk + 2Δx – 2Δy

}