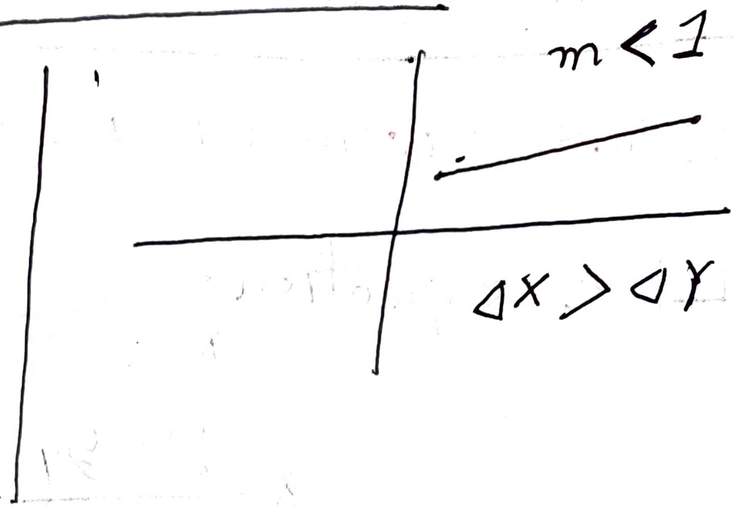
DDA Line Drawing Algorithm

When,

$m < 1$

$$x_{k+1} = x_k + 1$$

$$y_{k+1} = y_k + \frac{\Delta y}{\Delta x}$$
$$= y_k + m$$

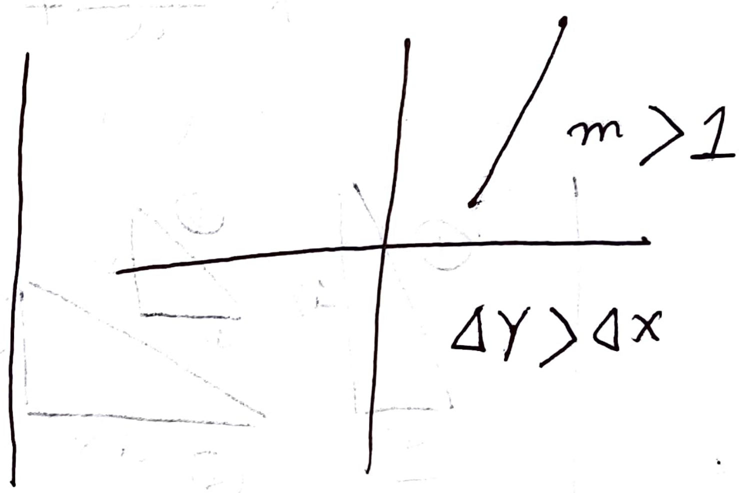


when,

$m > 1$

$$y_{k+1} = y_k + 1$$

$$x_{k+1} = x_k + \frac{\Delta x}{\Delta y}$$
$$= x_k + \frac{1}{m}$$

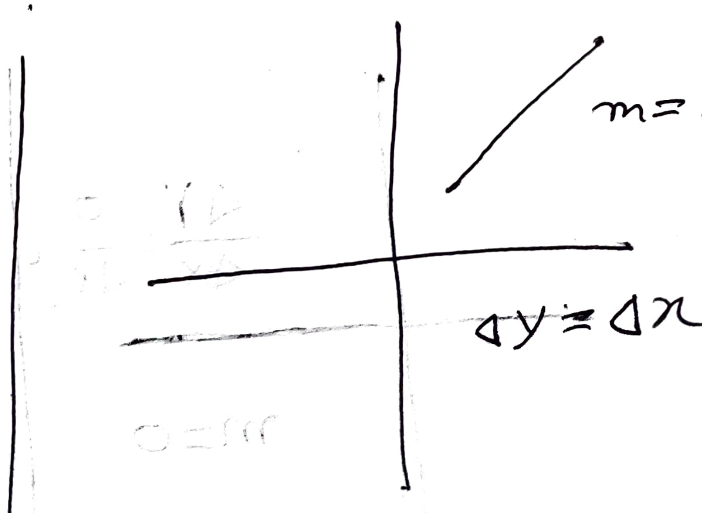


when,

$m = 1$

$$y_{k+1} = y_k + 1$$

$$x_{k+1} = x_k + 1$$



DDA Algorithm Sudo Code :

```
DDA (x1, y1, x2, y2)

dx = x2 - x1

dy = y2 - y1

if ( abs (dx) > abs(dy))
        steps = abs(dx)

else    steps = abs(dy)

x inc = dx/steps

y inc = dy/ steps

for( int i=1; k=steps; i++)

  {  putpixal (x1, y1);

     x1 = x1 + x inc

     y1 = y1 + y inc

  }
```

Bresenham Algorithm Sudo code:

```
Bresenham(x1, y1, x2, y2)
{ x = x1
  y = y1
  dx = x2 - x1
  dy = y2 - y1
  P = 2xdy - dx
While (x <= x2)
  { putpixel(x, y);
    x++;
    if (P<0)
        P = P + 2dy;
    else {
        P = P + 2dy - 2dx;
        y++;
    }
}
```

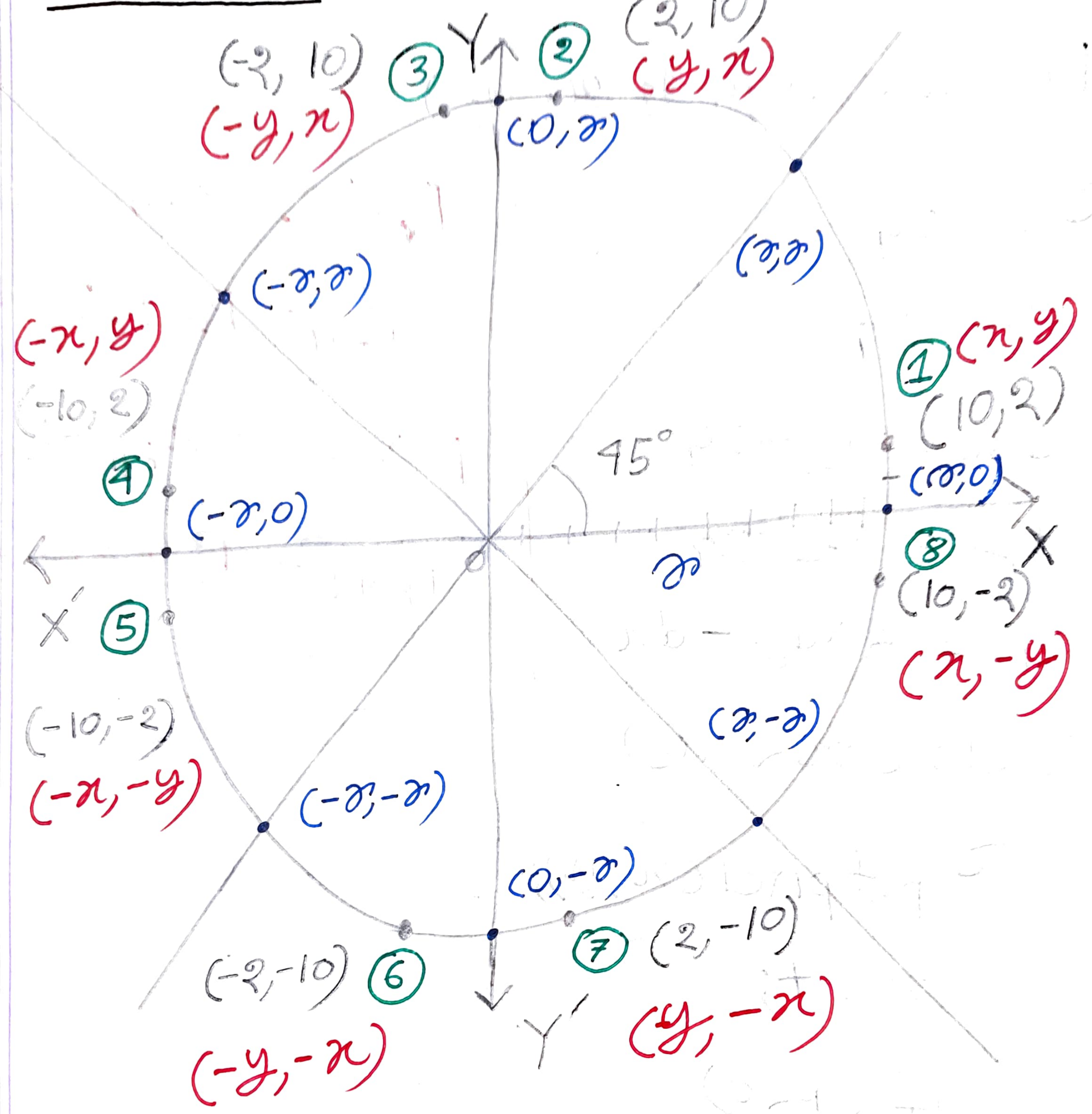
$P_k < 0$, (x_{k+1}, y_k)

$P_{k+1} = P_k + 2\Delta y$

$P > 0$, $(x_{k+1}, y_k$

$P_{k+1} = P_k + 2\Delta y$

A Circle: $(0,0), r \rightarrow x^2+y^2=r^2$



Circle: Set of points that lie at an equ
distance from a fixed point called
center.

Brute force approach for circle drawing:

way $x \rightarrow 1$ to r

$y = ?$ [We have to find y for all x]

$$x^2 + y^2 = r^2$$

$$y = \sqrt{r^2 - x^2}$$

Drawback:

⇒ For each step we have to find square root, which is very complex.

⇒ It will take much time & Computatio

Bresenham's Circle Drawing Algo. Sudo Code

circle (x, y) (0,0) → r

```
{
    x = 0;
    y = r;
    d = 3 - 2r
  while (x <= y)
    { set. pixel (x, y);
      if (d < 0)
        { d = d + 4x + 6}
        }
      else { d = d + 4(x - y) + 10
        y--;
        }
      x++;
}
```

Mid point circle drawing Algo Sudo code:

$(0,0) \rightarrow r$

```
circle (x, y)
{   x = 0;
    y = r;
    P = 5/4 - r = 1 - r
while ( x <= y)
  {
     setpixel( x, y);
   If (P < 0)
     {  p = p + 2x + 3;
     }
   else
     {  P = p + 2(x - y) + 5;
        y--;
     }
   x++;
}
```

B.L

$P_0 = 2\Delta y - \Delta x$

$P<0 \quad (x++, y), \quad P_{k+1} = P_k + 2\Delta y$

$P \geq 0 \quad (x++, y++), \quad P_{k+1} = P_k + 2\Delta y - 2\Delta x$

B.C

$d_0 = 3 - 2r$

$d<0, \ (x++, y), \ d_{i+1} = d_i + 4x_i + 6$

$d \geq 0, \ (x++, y--), \ d_{i+1} = d_i + 4(x_i - y_i) + 10$

Mid.C.

$P_0 = \frac{5}{4} - r$

$P<0, \ (x++, y), \ P_{i+1} = P_i + 2(x_i + 1) + 1$

$P \geq 0, \ (x++, y--) \ P_{i+1} = P_i + 2(x_i - y_i) + 5$

Flood-fill algo: Sudo code: 4 connected

```
Flood Fill (int x, int y, fill color, original_color)
{ int color;
  get pixel (x, y, color);
if ( color == original_color) {
  set pixel (x, y, fill_color);
  Flood Fill (x+1, y, fill_color, original_color)
   "     "   (x, y+1, fill_color, original_color);
   "     "   (x-1, y, fill_color; original_color);
   "     "   ( x, y-1, fill_color; original_color))

} }
```

Boundary-fill algo. subcode: 4 connected

```
Boundary-fill (int x, int y, fill color, boundary color)
{ int color;
  get pixel (x, y, color);
if ( color != boundary_color && color != fill_color
{ set Pixel (x, y, fill_color);
  Boundary-fill (x, y, fill_color, boundary_Colo
  - - - - - - - - -
}
```