"Point drawing"

way-1

$x, y \in R$

Given $P(x, y)$

$x' = floor(x)$

$y' = floor(y)$

putpixel → $(x', y', color)$

way-2

~~putpixel (x', y', color)~~

$x, y \in R$

Given, $P(m, y)$

$x' = floor(x + 0.5)$;

$y' = floor(y + 0.5)$;

putpixel $= f(x', y', color)$

"Line Drawing"

- → Direct Equation
- → DDA (Digital Differential Analyzer);
- → Bressenham line drawing algo.
- → Mid Point line drawing Algo.

Q What is line?

<u>Scan conversion</u>: A process of drawing geometric objects from it's geometric defination to a set of pixels by using computer graphics system.

<u>Scan convension;</u>

↳ point
line
circle
ellips
triangle
square
Filled polygon

$(x-h)^2 + (y-k)^2 = r^2$

$r = 0$

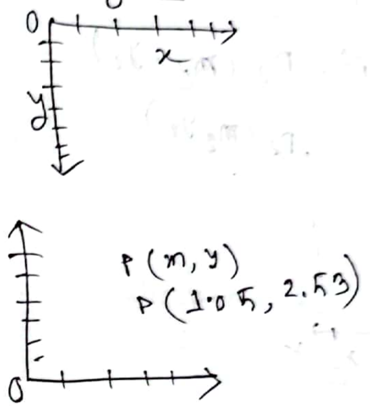
$r > 0$ circle

$y = mx + b$ (m → slope)

$m = \tan\theta$

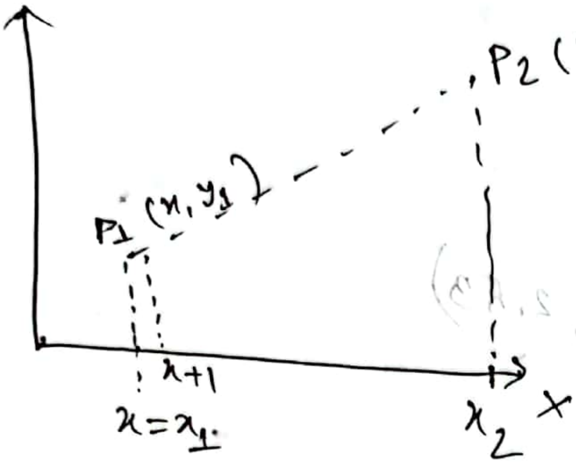
$m = \frac{y_2 - y_1}{x_2 - x_1}$

<u>Point drawing:</u>



Direct Equation

$P_1(x_1, y_1)$, $P_2(x_2, y_2)$



(i) $$m = \frac{y_2 - y_1}{x_2 - x_1}$$

(ii) $$b = y - mx$$
$$= y_1 - mx_1$$

$$y = mx + b$$

putpixel (x, y, color)

```
while (x < x2)
{
    x++;
    y = mx + b
    putpixel (m, y, color);
}
```

Algorithm:

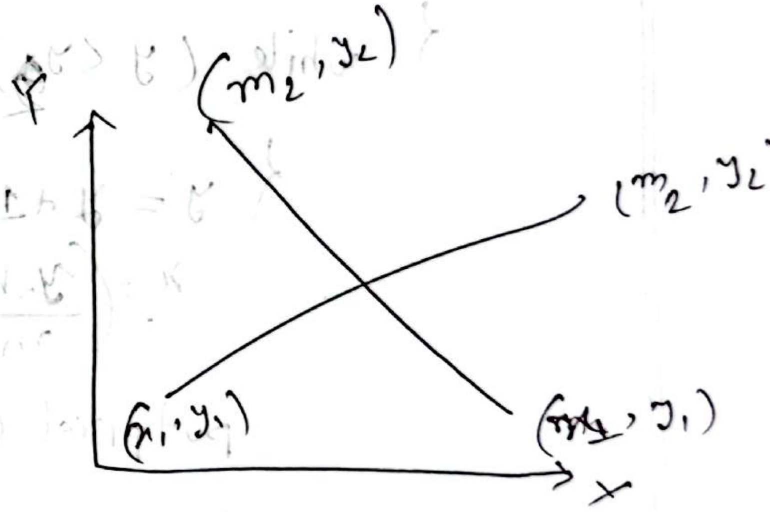
Input : $P_1(x_1, y_1)$, $P_2(x_2, y_2)$

Output : A line that is drawn by your CGS.

process:

(i) Inialization : calculate m & b.

$$m = \frac{y_2 - y_1}{x_2 - x_1}, \quad b = y - mx$$



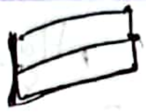
(ii) if $(x_1 < x_2)$

{ $x = x_1$;
  $y = y_1$;
}

else

{ $x = x_2$;
  $y = y_2$;
}

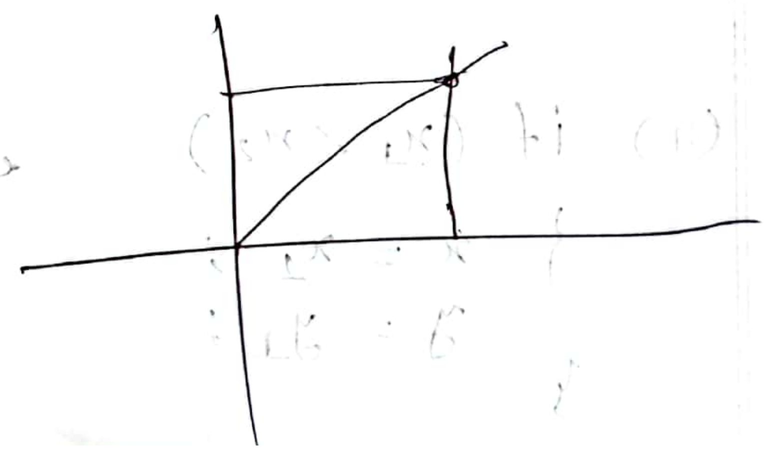
319, 0, 319, 479
0, 239, 639, 239



(iii) putpixel (x, y. clr)

(iv) if (|m| ≤ 1)

```
{ while (x < x2)
    { x = x+1;
      y = mx+b;
      putpixel (x, y, clr);
    }
}
else if (|m| > 1)
{ while (y < y2)
    { y = y+1;
      x = ((y-b)/m);
      putpixel (x, y, clr);
    }
}
```



Pros:

(1) easy to draw

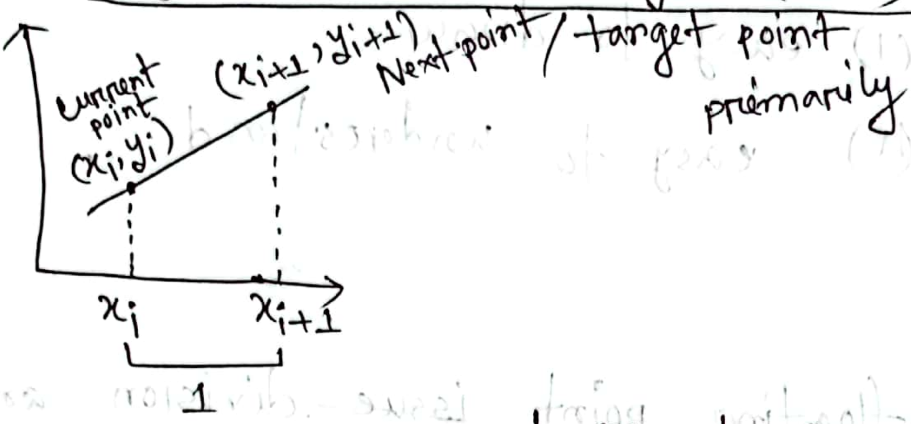
(ii) easy to understand.

cons:

(1) floating point ~~issue~~. division and multiplication.

"28.01.2025"

Digital Differential Analyzer (DDA)



$$m = \frac{\Delta y}{\Delta x}$$

$$\Delta y = y_{i+1} - y_i$$

$$\Delta x = x_{i+1} - x_i$$

if $(|m| \le 1)$

{

$\Delta x = 1$

$\Rightarrow m = \frac{\Delta y}{1}$

$\Rightarrow \Delta y = m$

$\Rightarrow y_{i+1} - y_i = m$

$\Rightarrow y_{i+1} = y_i + m$

$x_{i+1} = x_i + 1$

}

$m = \tan\theta$

$= \tan(45)$

$= 1$

x এর value 1 করে বাড়াবো y বের করবো, m যদি 1 থেকে বড় তাহলে উল্টো।

else if $(|m| > 1)$

{

$\Delta y = 1$

$m = \frac{1}{\Delta x}$

$\Rightarrow m = \frac{1}{x_{i+1} - x_i}$

$\Rightarrow x_{i+1} - x_i = \frac{1}{m}$

$\Rightarrow x_{i+1} = x_i + \frac{1}{m}$

$y_{i+1} = y_i + 1$

}

Final Algo :

Input : $P_1(x_1, y_1)$, $P_2(x_2, y_2)$, where $x_1 < x_2$

Output : A line by using 2 points.

Steps:

(i) Initialization:

$$x = x_1, \quad y = y_1$$

$$m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$$

(ii) if ($|m| \leq 1$)

```
{
    while (x <= x2)
    {
        putpixel (x, y, color);
        x++;
        y = y + m;
    }
}
```

color = 0–15
RED

(iii) else if ($|m| > 1$)

```
{
    while (y <= y2)
    {
        putpixel (x, y, color);
        y++;
        x = x + 1/m;
    }
}
```

CamScanner

$$\frac{400-6}{500-5} = \frac{394}{495} = 0.795$$

Pros :

(i) No need of floating point multiplication.

Cons :

(i) Still floating point addition exist.

5 marks **Math:** Draw a line using (5,6), (8,12) these points using DDA algorithm.

Solution:

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$= \frac{12-6}{8-5} = \frac{6}{3} = 2$$

$\therefore |m| > 1$

~~$y_{i+1} = y_i + 1$~~ $y_{i+1} = y_i + 1$

$$x_{i+1} = x_i + \frac{1}{m}$$

$$= x_i + \frac{1}{2}$$

$$= x_i + 0.5$$

x	y	(x, y)
5	6	(5, 6)
5.5	7	(6, 7)
6	8	(6, 8)
6.5	9	(7, 9)
7	10	(7, 10)
7.5	11	(8, 11)
8	12	(8, 12)

Math: Draw a line $(1,1)$, $(11,10)$ using DDA.

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{10-1}{11-1} = \frac{9}{10} = 0.9$$

$$|m| \leq 1$$

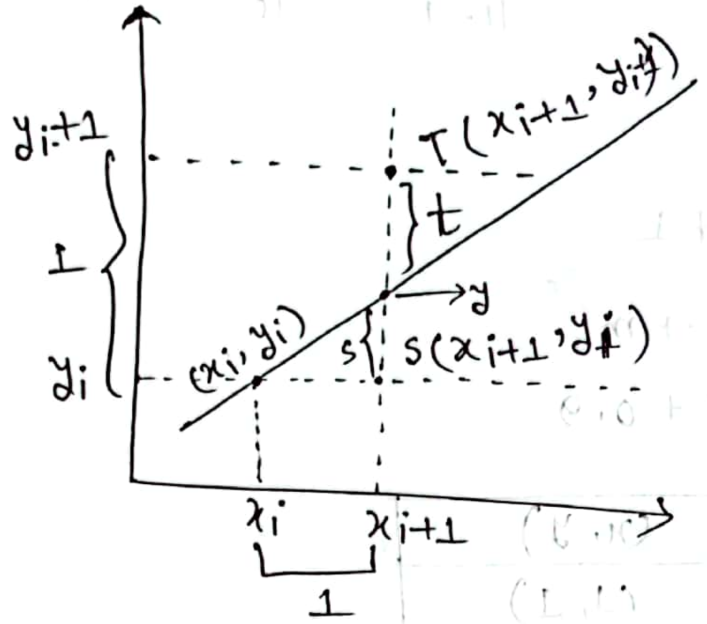
$$\therefore x_{i+1} = x_i + 1$$

$$y_{i+1} = y_i + m$$

$$= y_i + 0.9$$

x	y	(x, y)
1	1	(1, 1)
2	1.9	(2, 2)
3	2.8	(3, 3)
4	3.7	(4, 4)
5	4.6	(5, 5)
6	5.5	(6, 6)
7	6.4	(7, 6)
8	7.3	(8, 7)
9	8.2	(9, 8)
10	9.1	(10, 9)
11	10.0	(11, 10)

Brussenham Line drawing Algo :



$$s = y - y_i \quad \text{——— (i)}$$

$$t = y_i + 1 - y \quad \text{——— (ii)}$$

$$(s-t) = y - y_i - (y_i + 1 - y)$$

$$= y - y_i - y_i - 1 + y$$

$$= 2y - 2y_i - 1 \quad \text{——— (iii)}$$

Case : 1

$(s-t) < 0$

$\Rightarrow s < t$ (select point s)

For s :

$$x_{i+1} = x_i + 1$$

$$y_{i+1} = y_i$$

Case : 2

$(s-t) > 0$

$s > t$ (select point T)

For T :

$x_{i+1} = x_i + 1$

$y_{i+1} = y_i + 1$

From equation (iii)

$(s-t) = 2y - 2y_i - 1$

$\Rightarrow (s-t) = 2(mx_i + b) - 2y_i - 1$

$\Rightarrow (s-t) = 2\left(\frac{\Delta y}{\Delta x} x_i + b\right) - 2y_i - 1$

$\Rightarrow (s-t) = 2\left(\frac{\Delta y \cdot x_i + b\Delta x}{\Delta x}\right) - 2y_i - 1$

$\Rightarrow (s-t) = \frac{2x_i\Delta y + 2b\Delta x}{\Delta x} - 2y_i - 1$

$\Rightarrow \Delta x(s-t) = 2x_i\Delta y + 2b\Delta x - 2y_i\cancel{\Delta x} - \Delta x$

$\Rightarrow \Delta x(s-t) = 2x_i\Delta y + \underset{\text{constant}}{\underset{\downarrow}{2b\Delta x}} - 2y_i\Delta x - \underset{\text{constant}}{\underset{\downarrow}{\Delta x}}$

$\Rightarrow \Delta x(s-t) = 2x_i\Delta y - 2y_i\Delta x + \underbrace{\Delta x(2b-1)}_{\text{constant} = c}$

$\Rightarrow \Delta x(s-t) = 2x_i\Delta y - 2y_i\Delta x + c$ ——— (iv)

Let,

decision variable,

$$d_i = \Delta x\,(s-t)$$

$$\therefore d_i = 2x_i\Delta y - 2y_i\Delta x + c \quad \text{——— (V)}$$

$$\therefore d_{i+1} = 2x_{i+1}\Delta y - 2y_{i+1}\Delta x + c \quad \text{——— (VI)}$$

(VI) − (V) $\Rightarrow$

$$d_{i+1} - d_i = 2x_{i+1}\Delta y - 2x_i\Delta y - 2y_{i+1}\Delta x + 2y_i\Delta x + c$$

$$\Rightarrow d_{i+1} - d_i = 2\Delta y\,(x_{i+1} - x_i) - 2\Delta x\,(y_{i+1} - y_i) + c$$

$$\Rightarrow d_{i+1} = d_i + 2\Delta y\,(x_{i+1} - x_i) - 2\Delta x\,(y_{i+1} - y_i)$$

——— (VII)

For s :

$$x_{i+1} = x_i + 1$$

$$y_{i+1} = y_i$$

$$d_{i+1} = d_i + 2\Delta y\,(\cancel{x_i} + 1 - \cancel{x_i}) - 2\Delta x\,(y_i - y_i)$$

$$\Rightarrow d_{i+1} = d_i + 2\Delta y$$

$$\therefore d_{i+1} = d_i + 2\Delta y$$

For T:

$$x_{i+1} = x_i + 1$$
$$y_{i+1} = y_i + 1$$

$$d_{i+1} = d_i + 2\Delta y(\cancel{x_i} + 1 - \cancel{x_i}) - 2\Delta x(\cancel{y_i} + 1 - \cancel{y_i})$$

$$\Rightarrow d_{i+1} = d_i + 2\Delta y - 2\Delta x$$

$$\therefore d_{i+1} = d_i + 2\Delta y - 2\Delta x$$

$$d_{i+1} = \begin{cases} d_i + 2\Delta y, & S \text{ point } (d_i < 0) \\ d_i + 2\Delta y - 2\Delta x, & T \text{ point } (d_i \geq 0) \end{cases}$$

বাকিটা: Next Page →

"Lab- 2"

"29.01.2025"

x_1	y_1	x_2	y_2	Direct Equation.
1	1	100	100	

theory 29.01.2025

সূত্র $d_{i+1} = \begin{cases} d_i + 2\Delta y \,;\; S \text{ point } (d_i < 0) \\ d_i + 2(\Delta y - \Delta x) \,;\; T \text{ point } (d_i \geq 0) \end{cases}$

Initial Condition : $d_1 = ?$

$$d_i = \Delta x (s-t)$$

From equation (iii) $\Rightarrow$

$$(s-t) = 2y - 2y_i - 1$$

$\Rightarrow \Delta x (s-t) = 2y\Delta x - 2y_i \Delta x - \Delta x$ [Δx দ্বারা গুণ]

$\Rightarrow d_i = 2y\Delta x - 2y_i \Delta x - \Delta x$

$\Rightarrow d_i = \Delta x (2y - 2y_i - 1)$

$\Rightarrow d_i = \Delta x [2(mx_i + b) - 2y_i - 1]$

$\Rightarrow d_i = \Delta x [2\{m(x_i + 1) + b\} - 2y_i - 1]$

$\Rightarrow d_i = \Delta x [2(mx_i + m + b) - 2y_i - 1]$

$\Rightarrow d_1 = \Delta x [2(mx_1 + m + b) - 2y_1 - 1]$

$\Rightarrow d_1 = \Delta x [2(\underline{mx_1} + m + \underline{b} - y_1) - 1]$

$\Rightarrow d_1 = \Delta x [2(\cancel{y_1} + m - \cancel{y_1}) - 1]$

$\Rightarrow d_1 = \Delta x (2m - 1)$

$$\Rightarrow d_1 = \Delta x\left(2\frac{\Delta y}{\Delta x} - 1\right)$$

$$\Rightarrow d_1 = \cancel{\Delta x}\ \frac{2\Delta y - \Delta x}{\cancel{\Delta x}}$$

$$\therefore d_1 = 2\Delta y - \Delta x$$ মুখস্থ

Final Algorithm:

$0 \leq \theta \leq 45^\circ$

Input : $P_1(x_1, y_2)$, $P(x_2, y_2)$

Output : A line given by 2 points.

$$d_{i+1} = \begin{cases} d_i + \underbrace{2dy}_{ds} & \text{spoint} \\ d_i + \underbrace{2(dy - dx)}_{dt} & \text{t point} \end{cases}$$

Steps:

$dx = x_2 - x_1$

$dy = y_2 - y_1$

$d = 2dy - dx$

$ds = 2dy$

$dt = 2(dy - dx)$

$ds = 2dy$

$dt = 2(dy - dx)$

$x = x_1$, $y = y_1$

```
while (x <= x2)
{     putpixel (x, y, color);
          if (d < 0)
          {
               x++;
               d = d + ds;
          }
```

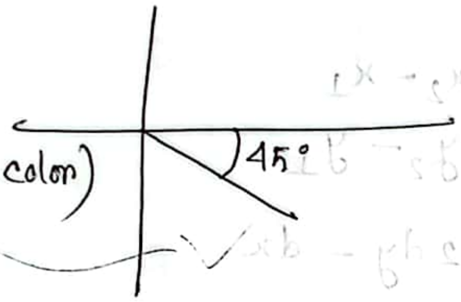
x = x1
y = y1

```
else {
        x++;
        y++;
        d = d + dt;
}   }
```

Pros:

(i) No floating point multiplication and addition is needed only intezere value.

যদি -45° হয়
তাহলে, putpixel (x, -y, color)



Math: $P_1(3, 6)$, $P_2(12, 13)$

Solution:

$dx = x_2 - x_1 = 12 - 3 = 9$

$dy = 13 - 6 = 7$

$d = 2dy - dx = 2 \cdot 7 - 9$
$= 14 - 9 = 5$

$ds = 2dy = 2 \cdot 7 = 14$

$dt = 2(dy - dx) = 2(7 - 9) = -4$

$x = 3, \; y = 6$

d	x	y	(x, y)
	3	6	(3, 6)
$5 \geq 0$	4	7	(4, 7)
$1 \geq 0$	5	8	(5, 8)
$-3 < 0$	6	8	(6, 8)
$11 \geq 0$	7	9	(7, 9)
$7 \geq 0$	8	10	(8, 10)
$3 \geq 0$	9	11	(9, 11)
$-1 < 0$	10	11	(10, 11)
$13 \geq 0$	11	12	(11, 12)
$9 \geq 0$	12	13	(12, 13)

Math: $P_1(1, 1)$, $P_2(50, 49)$

$$dx = x_2 - x_1 = 50 - 1 = 49$$

$$dy = y_2 - y_1 = 49 - 1 = 48$$

$$d = 2dy - dx = 2 \cdot 48 - 49$$
$$= 96 - 49 = 47.$$

$x = 1$

$y = 1$

$$ds = 2dy = 2 \cdot 48 = 96$$

$$dt = 2(dy - dx) = 2(48 - 49) = -2$$

d	x	y	(x, y)
	1	1	(1,1)
47≥0	2	2	(2,2)
45≥0	3	3	(3,3)
43≥0	4	4	(4,4)
41≥0	5	5	(5,5)
39≥0	6	6	(6,6)
37≥0	7	7	(7,7)
35≥0	8	8	(8,8)
33≥0	9	9	(9,9)
31≥0	10	10	(10,10)
29≥0	11	11	(11,11)
27≥0	12	12	(12,12)
25≥0	13	13	(13,13)
23≥0	14	14	(14,14)
21≥0	15	15	(15,15)
19≥0	16	16	(16,16)
17≥0	17	17	(17,17)
15≥0	18	18	(18,18)
13≥0	19	19	(19,19)
11≥0	20	20	(20,20)
9≥0	21	21	(21,21)
7≥0	22	22	(22,22)
5≥0	23	23	(23,23)
3≥0	24	24	(24,24)

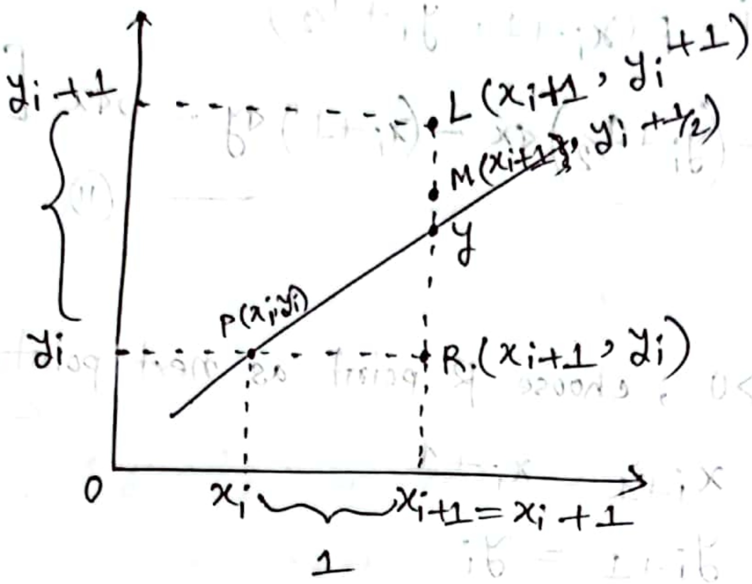
d	x	y	(x, y)
$1 \geq 0$	25	25	(25, 25)
$-1 < 0$	26	25	(26, 25)
$95 \geq 0$	27	26	(27, 26)
$93 \geq 0$	28	27	(28, 27)
$91 \geq 0$	29	28	(29, 28)
$89 \geq 0$	30	29	(30, 29)
$87 \geq 0$	31	30	(31, 30)
$85 \geq 0$	32	31	(32, 30)
$83 \geq 0$	33	32	(33, 32)
$81 \geq 0$	34	33	(33, 34)
$79 \geq 0$	35	34	(35, 34)
$77 \geq 0$	36	35	(36, 35)
$75 \geq 0$	37	36	(37, 35)
$73 \geq 0$	38	37	(38, 37)
$71 \geq 0$	39	38	(39, 38)
$69 \geq 0$	40	39	(40, 39)
$67 \geq 0$	41	40	(41, 40)
$65 \geq 0$	42	41	(42, 41)
$63 \geq 0$	43	42	(43, 42)
$61 \geq 0$	44	43	(44, 43)
$59 \geq 0$	45	44	(45, 44)

d	x	y	(x, y)
$57 \geq 0$	46	45	(46, 45)
$55 \geq 0$	47	46	(47, 46)
$53 \geq 0$	48	47	(48, 47)
$51 \geq 0$	49	48	(49, 48)
$49 \geq 0$	50	49	(50, 49)

29.01.25

" 04.02.202

" Mid Point line drawing "



We know,

$$y = mx + b$$

$$\Rightarrow y = \frac{\Delta y}{\Delta x} x + b$$

$$\Rightarrow \Delta x y = \Delta y x + b\Delta x$$

$$\Rightarrow y\Delta x - x\Delta y - b\Delta x = 0$$

let, $F(x,y) = y\Delta x - x\Delta y - b\Delta x$ ——— (1)

if, $F(x, y) = 0$; then, (x, y) on the line.

$F(x, y) < 0$; then (x, y) above the line.

$F(x, y) > 0$; then (x, y) below the line.

Let,

$$d_i = F(M \text{ point})$$ (d_i = decision variable)

$$\Rightarrow d_i = F(x_{i+1}, y_i + 1/2)$$

$$\Rightarrow d_i = F(x_i + 1, y_i + 1/2)$$

$$\therefore d_i = (y_i + 1/2)\Delta x - (x_i + 1)\Delta y - b\Delta x \quad \text{— (ii)}$$ [From equation (i)]

Case 1 :

$d_i > 0$; choose R point as next point.

$$x_{i+1} = x_i + 1$$

$$y_{i+1} = y_i$$

$$\therefore d_{i+1} = (y_{i+1} + 1/2)\Delta x - (x_{i+1} + 1)\Delta y - b\Delta x$$

$$= (y_i + 1/2)\Delta x - (x_i + 1 + 1)\Delta y - b\Delta x$$

$$= (y_i + 1/2)\Delta x - (x_i + 2)\Delta y - b\Delta x \quad \text{— (iii)}$$

(iii) − (ii) $\Rightarrow$

$$d_{i+1} - d_i = (y_i + 1/2 - y_i - 1/2)\Delta x - (x_i + 2 - x_i - 1)\Delta y - b\Delta x + b\Delta x$$

$$\Rightarrow d_{i+1} - d_i = 0 - \Delta y + 0$$

$$\Rightarrow d_{i+1} - d_i = -\Delta y$$

$\Rightarrow d_{i+1} = d_i - \Delta y$

$\therefore d_{i+1} = d_i - \Delta y$ সূত্রম্

Case : 2

$d_i < 0$; choose L point as next point.

$x_{i+1} = x_i + 1$

$y_{i+1} = y_i + 1$

$$\therefore d_{i+1} = \left(y_{i+1} + \tfrac{1}{2}\right)\Delta x - (x_{i+1} + 1)\Delta y - b\Delta x$$

$$= \left(y_i + 1 + \tfrac{1}{2}\right)\Delta x - (x_i + 1 + 1)\Delta y - b\Delta x$$

$$= \left(y_i + \tfrac{3}{2}\right)\Delta x - (x_i + 2)\Delta y - b\Delta x \quad \text{--- (iv)}$$

(iv) − (ii) $\Rightarrow$

$$d_{i+1} - d_i = \left(\cancel{y_i} + \tfrac{3}{2} - \cancel{y_i} - \tfrac{1}{2}\right)\Delta x - (\cancel{x_i} + 2 - \cancel{x_i} - 1)\Delta y - b\Delta x + b\Delta x$$

$$\Rightarrow d_{i+1} - d_i = \Delta x - \Delta y + 0$$

$$\Rightarrow d_{i+1} - d_i = \Delta x - \Delta y$$

$\therefore d_{i+1} = d_i + \Delta x - \Delta y$ সূত্রম্

$$d_{i+1} = \begin{cases} d_i - \Delta y \; ; \; (d_i \geq 0), \text{ R point} \\ d_i + \Delta x - \Delta y \; ; \; (d_i \leq 0), \text{ L point} \end{cases}$$

Initial Condition :

From equation (ii) $\Rightarrow$

$$d_i = \left(y_i + \tfrac{1}{2}\right)\Delta x - (x_i + 1)\Delta y - b\Delta x$$

$$\therefore d_1 = \left(y_1 + \tfrac{1}{2}\right)\Delta x - (x_1 + 1)\Delta y - b\Delta x$$

$$= y_1 \Delta x + \tfrac{1}{2}\Delta x - x_1 \Delta y - \Delta y - b\Delta x$$

$$= F(x_1, y_1) + \tfrac{1}{2}\Delta x - \Delta y \quad [\text{From equation (i)}]$$

Hence ; (x_1, y_1) on the line ;

$$\therefore F(x_1, y_1) = 0$$

$$\Rightarrow d_1 = 0 + \tfrac{1}{2}\Delta x - \Delta y$$

$$\therefore d_1 = \tfrac{1}{2}\Delta x - \Delta y$$ জরুরি

Final Algorithm

Input : $P_1(x_1, y_1)$ & $P_2(x_2, y_2)$

Output : A line given by 2 points.

Steps :

$$d_{i+1} = \begin{cases} d_i - \underbrace{\Delta y}_{dR} ; & (d_i > 0) \text{ R point} \\ d_i + \underbrace{\Delta x - \Delta y}_{dL} ; & (d_i < 0) \text{ L point} \end{cases}$$

$$dx = x_2 - x_1$$

$$dy = y_2 - y_1$$

$$d_1 = \tfrac{1}{2} dx - dy$$

$\therefore dR = -dy$

$\therefore dL = dx - dy$

$x = x_1, \; y = y_1$

```
while (x <= x2)
{
    putpixel (x, y, color);
    if (d >= 0)
    {
        x++;
        d = d + dR
    }
    else
    {
        x++;
        y++;
        d = d + dL;
    }
}
```

Math: $P_1(20, 10)$, $P_2(30, 18)$ draw line using Mid Point Algo.

Solution:

$dx = 30 - 20 = 10$

$dy = 18 - 10 = 8$

$d_1 = \frac{1}{2} dx - dy = \frac{10}{2} - 8 = 5 - 8 = -3$

$\therefore dR = -8$

$\therefore dL = 10 - 8 = 2$

$x = 20$

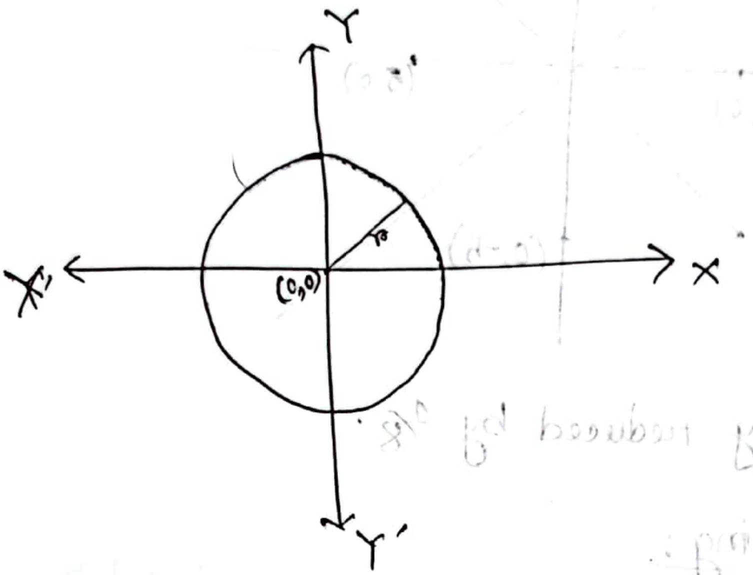
$y = 10$

d	x	y	(x, y)
	20	10	(20, 10)
-3<0	21	11	(21, 11)
-1<0	22	12	(22, 12)
1≥0	23	12	(23, 12)
-7<0	24	13	(24, 13)
-5<0	25	14	(25, 14)
-3<0	26	15	(26, 15)
-1<0	27	16	(27, 16)
1≥0	28	16	(28, 16)
-7<0	29	17	(29, 17)
-5<0	30	18	(30, 18)

Implement DDA line drawing Algo $P_1(5, 6)$, $P_2(500, 400)$

"Theory" "05.02.2025"

"Circle drawing"



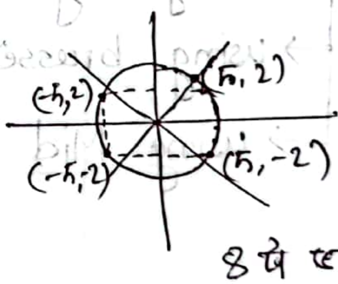
$x^2 + y^2 = r^2 \longrightarrow$ মূলবিন্দুগামী circle

$(x-h)^2 + (y-k)^2 = r^2$

Interesting Property:

→ 8 point symmetry

→ Reflecting property



8 টা ভাগ

If $x = 5, y = 2$

$P_1(x, y) = (5, 2)$

$P_2(x, -y) = (5, -2)$

$P_3(y, x) = (2, 5)$

$P_4(y, -x) = (2, -5)$

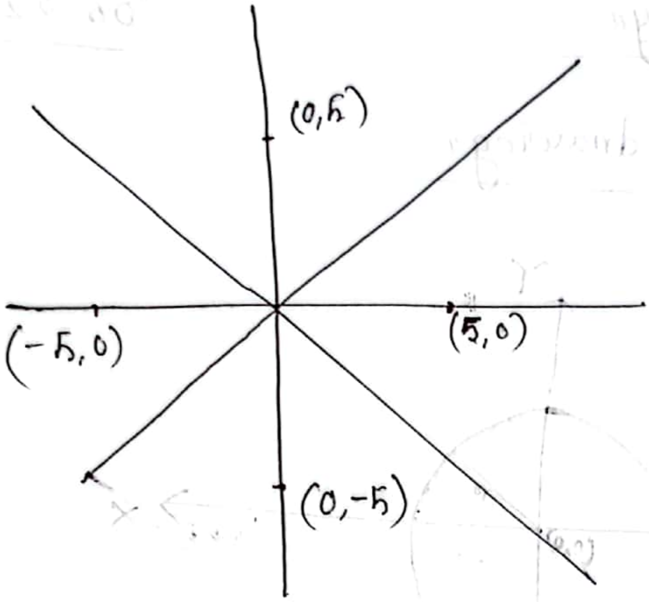
$P_5 = (-x, y) = (-5, 2)$

$P_6(-x, -y) = (-5, -2)$

$P_7(-y, x) = (-2, 5)$

$P_8(-y, -x) = (-2, -5)$

Radius ⇒ $r = 5$



Pros :

Complexity reduced by 1/8.

Circle drawing :

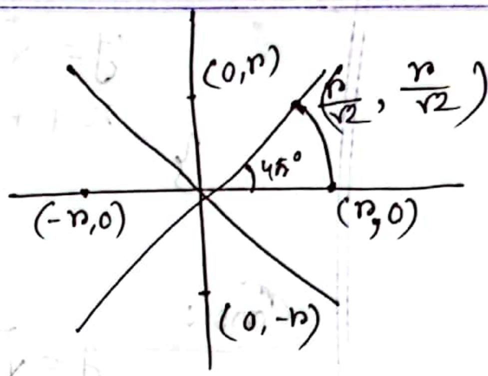
→ using direct equation
→ using trigonometric function
→ using bressenham circle drawing
→ using Mid Point Circle drawing.

"Direct Equation"

Circle equation, $x^2+y^2=r^2$

$x = r\cos\theta$

$y = r\sin\theta$



$x = r\cos 45°$

$= r.\frac{1}{\sqrt{2}} = \frac{r}{\sqrt{2}}$

$y = r\sin 45°$

$= r.\frac{1}{\sqrt{2}} = \frac{r}{\sqrt{2}}$

$x = r\cos 0°$

$= r$

$y = r\sin 0° = 0$

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

$\Rightarrow y^2 = r^2 - x^2$

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

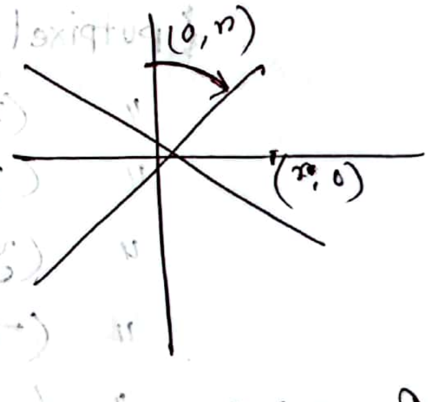
Given,

Centre (0, 0)

Radius = R

$x = 0$

$y = r$



while $(x <= \frac{r}{\sqrt{2}})$

{

putpixel (x, y, color);

putpixel (x, -y, color);

putpixe(y, x, clr);

putpixe (y, -x, clr);

putpixel (-x, y, clr);

putpixe(-x, -y, color);

putpixe (-y, x, color);

putpixel (-y, -x, color);

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

$x{+}{+};$

}

Cons :

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

এখানে root করলে y এর মান float আসবে কিন্তু pixel -এ int তাই এটা সমস্যা।

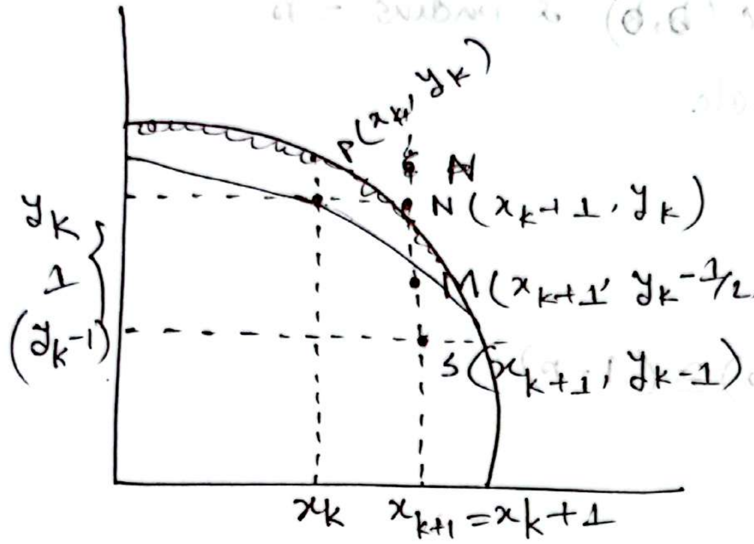
Trigonometric Approach:

```
θ = 0°
while (θ <= 45°)
  {
    x = r cos θ ;
    y = r sin θ ;
    { putpixel (x, y, clr);
        "       (x, -y, clr);
        "       (y, x, clr);
        "       (y, -x, clr);
        "       (-x, y, clr);
        "       (-x, -y, clr);
        "       (-y, x, clr);
        "       (-y, -x, clr);
  }  θ = θ + 1;
```

square root নাই কিন্তু floating point আছে।

"Mid Point Circle Drawing"

11.02.2025



If m inside the circle then choose N point

" m outside " " " " S Point

$P_k \geq 0 \longrightarrow$ M outside the circle (choose S point)

$P_k < 0 \longrightarrow$ M inside the circle (choose N point)

$$P_{k+1} = \begin{cases} P_k + 2(x_k - y_k) + 5 \; ; & P_k \geq 0 \; ; \text{ S point} \\ P_k + 2x_k + 3 \; ; & P_k < 0 \; ; \text{ N point} \end{cases}$$

S point $\equiv S(x_k + 1, \; y_k - 1)$

N point $\equiv N(x_k + 1, \; y_k)$

$$P_0 = \left(\frac{5}{4} - r\right) \cong (1 - r)$$

Lab ← $\left(\frac{5}{4} - r\right)$; $(1 - r)$ → math

$x = 0$

$y = r$

$P_k = 1 - r$

$x \le y$

$P \geq 0$

$P = P + 2(x-y) + 5$

$x++;$

$y--;$

$P < 0$

$P = P + 2x + 3$

$x++$

(0, 0) & ~~10~~ radius 10

$P = 1 - 10 = -9$

P	x	y	(x, y)
	0	10	(0,10)
-9 < 0	1	10	(1,10)
-12 < 0	2	10	(2,10)
-13 < 0	3	10	(3,10)
-12 < 0	4	10	(4,10)
-9 < 0	5	10	(5,10)
-4 < 0	6	10	(6,10)
3 ≥ 0	7	9	(7,9)
0	7	8	(7,8)
1 > 0	7	7	(7,7)
4 ≥ 0	7	6	(7,6)
9 ≥ 0			

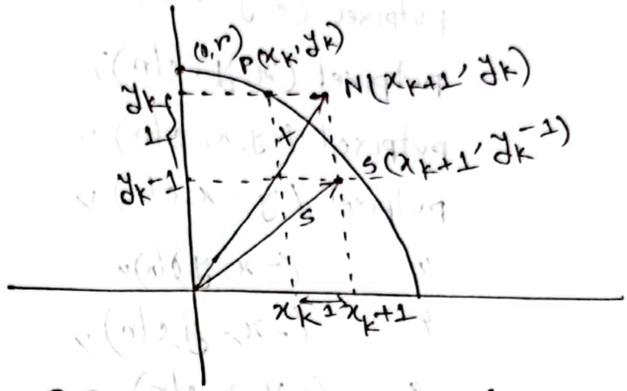
$(4, 5)$ .. $r = 15$

$x = 0,\ r = 15,\quad P = 1 - r = 1 - 15 = -14$

P	x	y	(x, y)
	0	15	(4, 20)
$-14 < 0$	1	15	(5, 20)
$-11 < 0$	2	15	(6, 20)
$-6 < 0$	3	15	(7, 20)
$1 \geq 0$	4	14	(8, 19)
$-18 < 0$	5	14	(9, 19)
$-7 < 0$	6	14	(10, 19)
$6 \geq 0$	7	13	(11, 18)
$-5 < 0$	8	13	(12, 18)
$12 \geq 0$	9	12	(13, 17)
$7 \geq 0$	10	11	(14, 16)
$6 \geq 0$	11	10	(15, 15) X
$9 \geq 0$			

"12.02.2025"

"Bressenham Circle Drawing Algorithm"



decision variable,

$$P_{K+1} = \begin{cases} P_K + 4x_K + 6 \ ; \ P_K < 0 \ ; (N \text{ point}) \\ P_K + 4(x_K - y_K) + 10 \ ; \ P_K \geq 0 \ ; (S \text{ point}) \end{cases}$$

Initial Condition :

$$P_0 = (3 - 2r)$$

Final Algo:

Input : center c(0, 0), radius = r.

Output : A circle.

Steps:

$x = 0, \ y = r$

$P = 3 - 2r$

```
while (x <= y)
{
      putpixel (x, y, color);
      putpixel (x, -y, clr);
      putpixel (y, x, clr);
      putpixel (y, -x, clr);
         "    (-x, y, clr);
         "    (-x, -y, clr);
         "    (-y, x, clr);
         "    (-y, -x, clr);
   if (P>=0)
   {
      P = P + 4(x - y) + 10;
      x++;
      y--;
   }
   else
   {
      P = P + 4x + 6;
      x++;
   }
}
```

* Draw a circle with center (0,0) and radius 1
using bressenham circle drawing algorithm.

Solution:

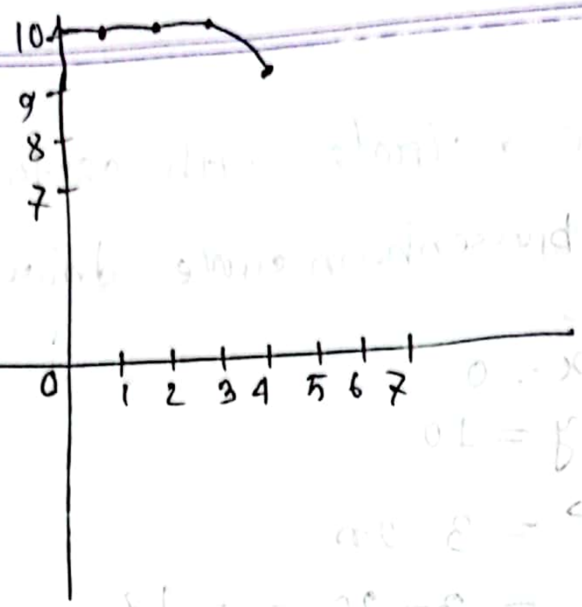
$x = 0$

$y = 10$

$P = 3 - 2r$

$= 3 - 20 = -17.$

P	x	y	(x, y)
	0	10	(0, 10)
$-17 < 0$	1	10	(1, 10)
$-11 < 0$	2	10	(2, 10)
$-1 < 0$	3	10	(3, 10)
$13 \geq 0$	4	9	(4, 9)
$-5 < 0$	5	9	(5, 9)
$17 \geq 0$	6	8	(6, 8)
$11 \geq 0$	7	7	(7, 7)
$13 \geq 0$	8	6	(8, 6)



$-27+4\times0+6$

* Draw a circle with center (5, 6) and radius 15.

$x = 0$

$y = 15$

$P = 3 - 2r$

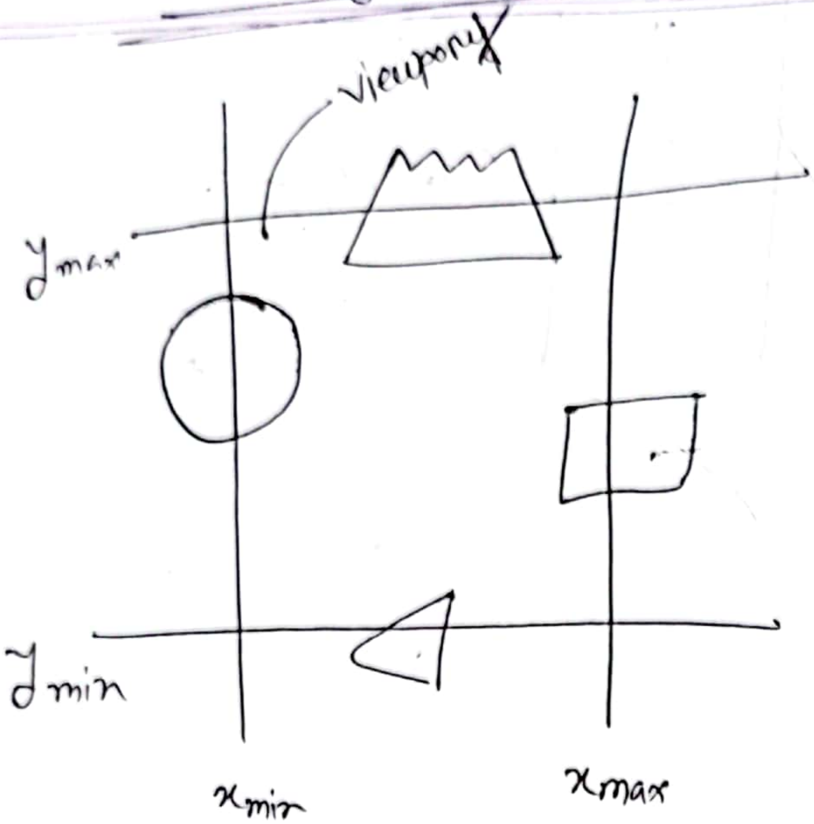
$= 3 - 30 = -27$

P	x	y	(x, y)
	0	15.	(5, 21)
$-27<0$	1	15	(6, 21)
$-21<0$	2	15	(7, 21)
$-11<0$	3	15	(8, 21)
$3\geq0$	4	14	(9, 20)
$-35<0$	5	14	(10, 20)
$-13<0$	6	14	(11, 20)
$13\geq0$	7	13	(12, 19)
$-9<0$	8	13	(13, 19)
$25\geq0$	9	12	(14, 18)
$15\geq0$	10	11	(15, 17)

P	x	y	(x, y)
13>,0	11	10	(16, 16)
19>,0	~~10~~		

19.02.2025

"Clipping"



Point clipping:

Visible:

$x_{min} \leq x \leq x_{max}$

$y_{min} \leq y \leq y_{max}$

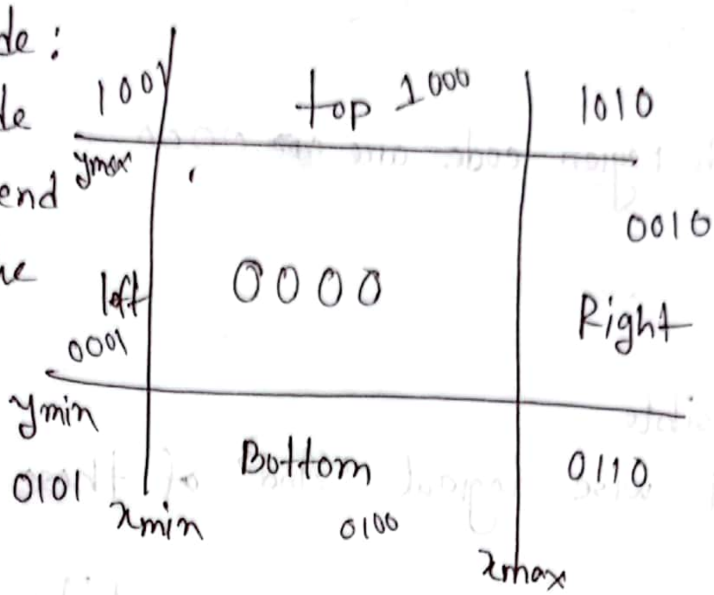
Line Clipping:

- → Cohen Sutherland.
- → Liang Barsky.

Cohen Sutherland:

Region code:

→ 4 bit code for each end point of the line.



Bit 1: If a point is in top then 1
otherwise 0

Bit 2: If a point is bottom then 1
otherwise 0

Bit 3: If a point is Right then 1
otherwise 0

Bit 4: If a point is in left then 1
otherwise 0

Condition on visible or not :

(i) Visible

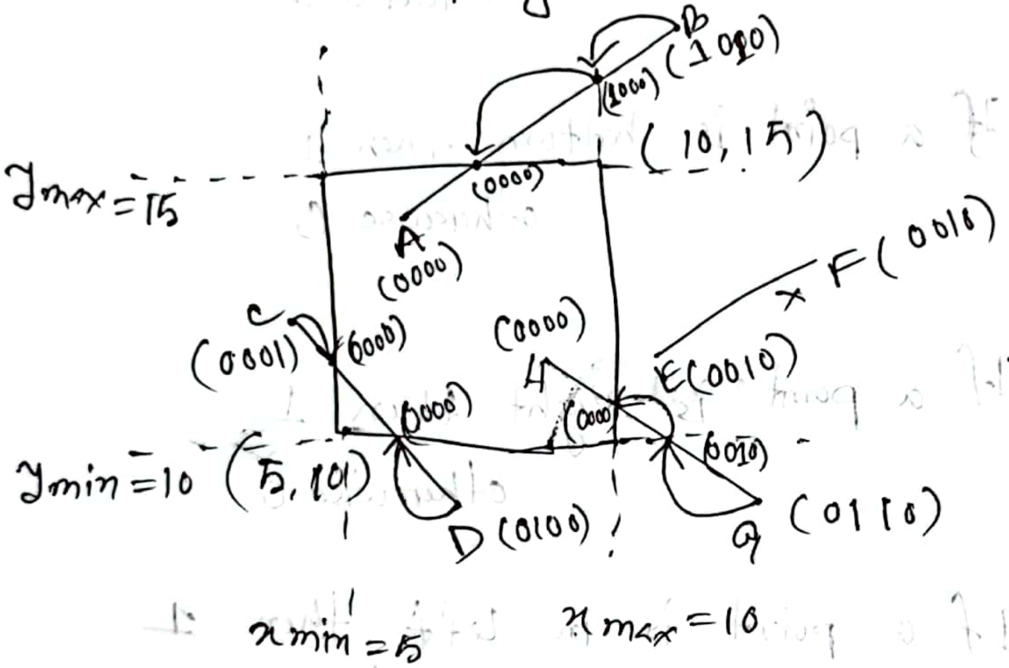
→ Both region code are .0000 .

(ii) Not visible

→ Bit wise logical "And" of these 2 region codes.

→ If not zero, then completly outside

→ If zero, clipping candidate.



একটা point - এর জন্য সর্বোচ্চ ২ টা step
" line " " " " "

Clipping candidate: $(x, y) = (x', y') = ?$

If
$$\left.\begin{array}{l} \text{Bit } 1 = 1, \; y' = y_{max} \\ \text{Bit } 2 = 1, \; y' = y_{min} \end{array}\right\} x' = ?$$

$$\left.\begin{array}{l} \text{Bit } 3 = 1, \; x' = x_{max} \\ \text{Bit } 4 = 1, \; x' = x_{min} \end{array}\right\} y' = ?$$

$$y = mx + c \quad \text{—— (i)}$$

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$y_1 = mx_1 + c \quad \text{—— (ii)}$$

$$y_2 = mx_1 = c \quad \text{—— (iii)}$$

(i) $\Rightarrow$

$$y' = mx' + c$$

$$= mx' + y_1 - mx_1$$

$$\therefore y' = y_1 + m(x' - x_1)$$

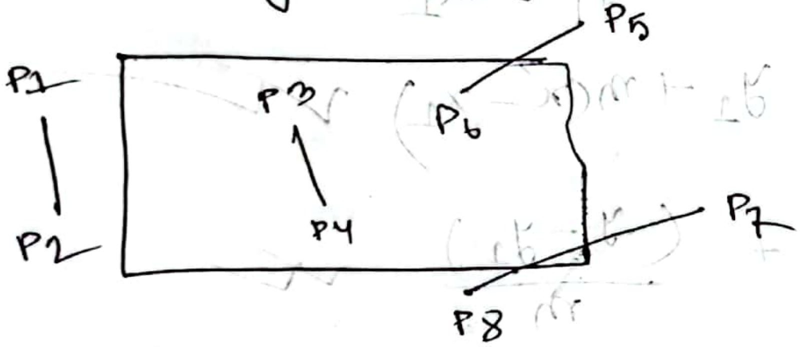
$$x' = x_1 + \frac{(y' - y_1)}{m}$$

"Cohen Sutherland Algo"

Step :

1. Find the region code
2. If both end point is 0000 then accept the line (completly visible)
3. Else perform the logical "and" operation.
   3.1. If result is not 0000 then reject the line.
   3.2. Else clip the line.
4. Replace with new region code.
5. Repeat the step 2.

Clip the following Line using Cohen Sutherland Algo.



For line $P_1 P_2$

$P_1 \rightarrow 0001$

& $P_2 \rightarrow 0001$

$0001 \rightarrow$ Not zero, so completly outside (Not visible)

For line P_3P_4 :

$P_3 = 0000 \longrightarrow$ zero
$P_4 = 0000 \longrightarrow$ zero

$0000 \longrightarrow$ Accept the line (Completly visible)

For line P_5P_6 :

$P_5 = 1000 \longrightarrow$ not zero
& $P_6 = 0000 \longrightarrow$ zero

$0000 \longrightarrow$ ~~Not~~ zero, clipping candidate.

$P_5' = 0000 \longrightarrow$ zero
& $P_6 = 0000 \longrightarrow$ zero

$0000 \longrightarrow$ zero, completely visible.

For Line P_7P_8 :

$P_7 = 0010 \longrightarrow$ not zero
& $P_8 = 0100 \longrightarrow$ not zero

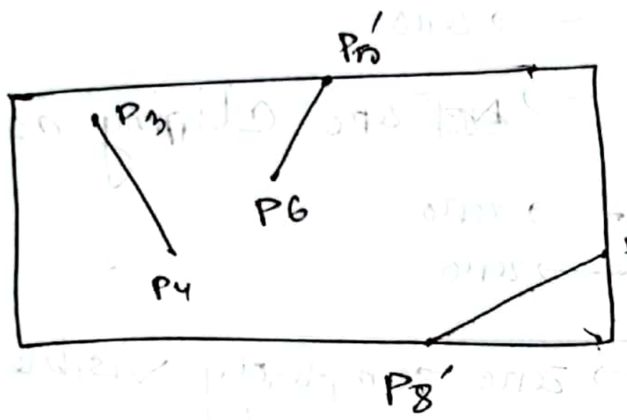
$0000 \longrightarrow$ zero, clipping candidate,

$P_7' = 0000 \rightarrow$ zero

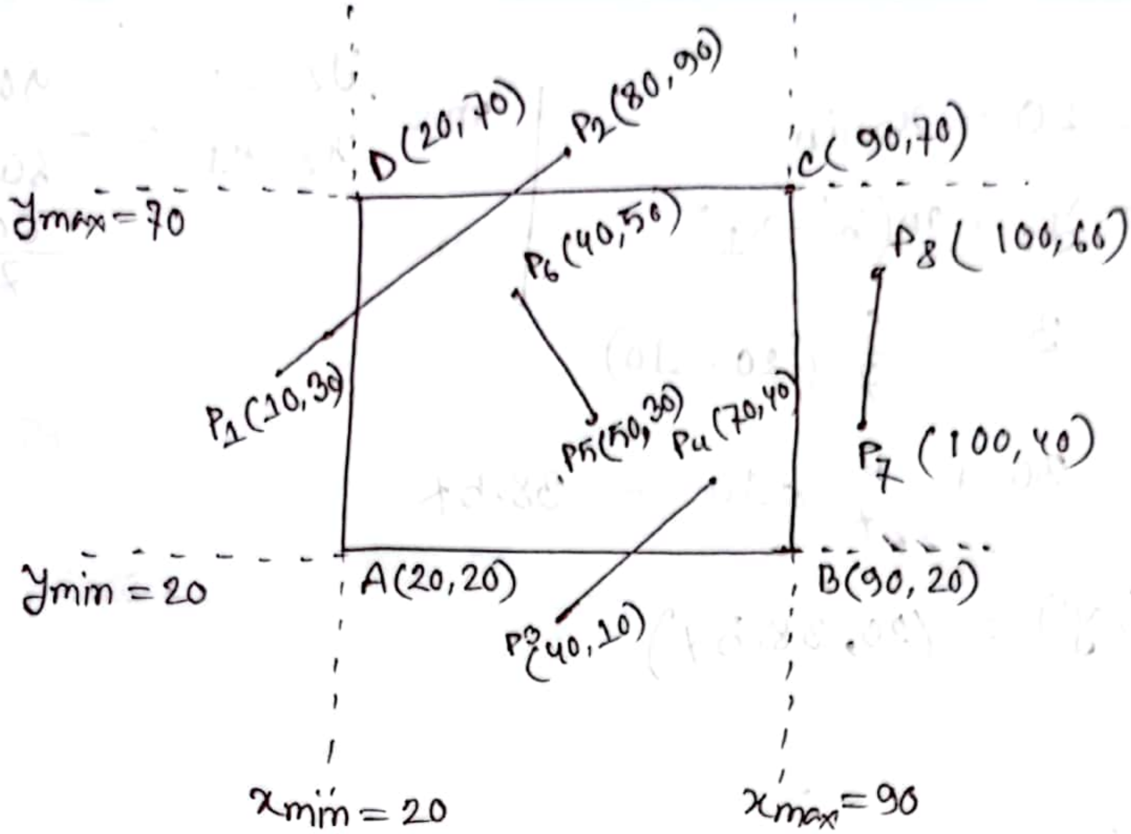
& $P_8 = 0100 \rightarrow$ not zero

$0000 \rightarrow$ zero ~~som~~ clipping candidate.

$P_7' = 0000 \rightarrow$ zero
$P_8' = 0000 \rightarrow$ zero
} completely visible :



final output



Q. Find the clipped line using cohen sutherland Algo.

Solution :

For P_1P_2 :

$P_1 = 0001 \longrightarrow$ not zero

& $P_2 = 1000 \longrightarrow$ not zero

$0000 \rightarrow$ zero, hence clipping candidate.

$P_1' = 0000 \longrightarrow$ zero

& $P_2 = 1000 \longrightarrow$ not zero

$0000 \rightarrow$ zero, hence clipping candidate.

$P_1' = 0000 \longrightarrow$ zero

& $P_2' = 0000 \longrightarrow$ zero

$0000 \longrightarrow$ zero, ful completely visible.

$P_1' \Rightarrow$

$$x' = 20 = x_{min}$$

$$y' = y_1 + m(x' - x_1)$$

$$= 30 + \frac{6}{7}(20 - 10)$$

$$= 30 + \frac{6}{7} \times 10 = 38.57$$

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{90 - 30}{80 - 10} = \frac{60}{70} = \frac{6}{7}$$

$\therefore P_1'(x', y') \equiv (20, 38.57)$

$P_2' \Rightarrow$

$$y' = 70$$

$$x' = x_1 + \frac{(y' - y_1)}{m}$$

$$= 80 + \frac{70 - 90}{6/7}$$

$$= 80 + \frac{\cancel{40 \times 7} - 20 \times 7}{6}$$

$$= \cancel{56.67}\ 80 - \frac{20 \times 7}{6}$$

$$= 56.67$$

$\therefore P_2'(x', y') \equiv (56.67, 70)$

For P_3P_4:

$P_3 = 0100 \longrightarrow$ not zero

& $P_4 = 0000 \longrightarrow$ zero

$0000 \longrightarrow$ zero, clipping candidate

$P_3' = 0000 \longrightarrow$ zero

$\& \; P_4 = 0000 \longrightarrow$ zero

$0000 \longrightarrow$ zero, completely visible.

$P_3' \Rightarrow$

$y' = 20$

$x' = x_1 + \dfrac{y' - y_1}{m}$

$= 40 + \dfrac{20 - 10}{1}$

$= 40 + 10 = 50$

$m = \dfrac{y_2 - y_1}{x_2 - x_1} = \dfrac{40 - 10}{70 - 40} =$

$\therefore P_3'(x', y') = (50, 20)$

For P_5P_6:

$P_5 = 0000 \longrightarrow$ zero

$\& \; P_6 = 0000 \longrightarrow$ zero

$0000 \longrightarrow$ zero, completely visible.

For P_7P_8:

$P_7 = 0010 \longrightarrow$ not zero

$\& \; P_8 = 0010 \longrightarrow$ not "

$0010 \longrightarrow$ not zero, fully outside.

"Syllabus"

→ Introduction (3 slides)

→ Line drawing
- → Direct
- → DDA
- → Bresenham
- → Mid Point

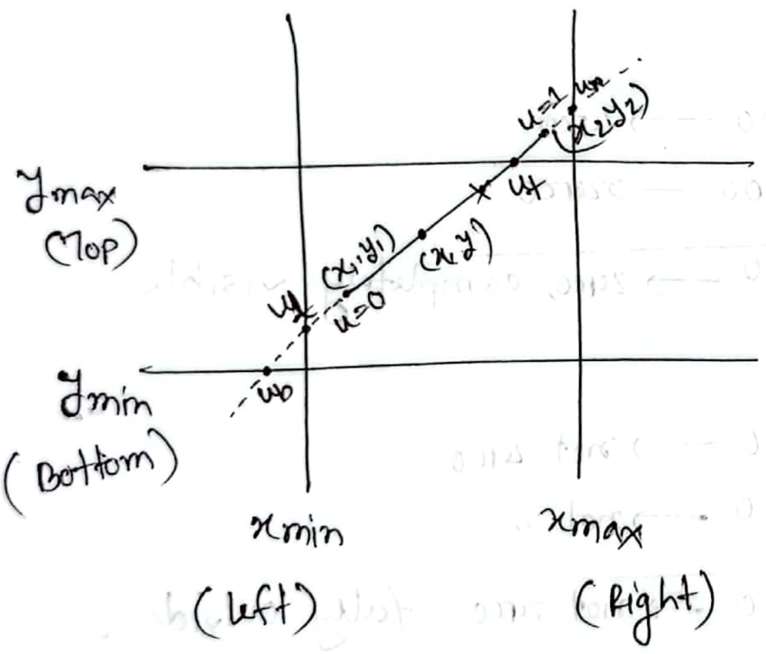
derivation

Math

→ Circle drawing (math)
- → Mid Point

"Liang Barsky Line Clipping Algo"

(Intercept point based)



$u = 0.2$

$0 \le u \le 1$

$x = x_1 + \Delta x\, u \qquad \Delta x = x_2 - x_1$

$y = y_1 + \Delta y\, u \qquad \Delta y = y_2 - y_1$

$u=0, \quad x=x_1, \; y=y_1$

$u=1, \quad x=x_2, \; y=y_2$

$u=0.5 \longrightarrow$ middle point

$x_{min} \leq x \leq x_{max}$ &&

$y_{min} \leq y \leq y_{max}$

$\Rightarrow x_{min} \leq x_1 + \Delta x u \leq x_{max}$ && $y_{min} \leq y_1 + \Delta y u \leq y_{max}$

$\Rightarrow p_k \cdot u \leq q_k, \quad k = 1, 2, 3, 4$

$\therefore p_1 = -\Delta x$ (left)

$\therefore q_1 = x_1 - x_{min}$ (left)

$p_2 = \Delta x, \; q_2 = x_{max} - x_1$ (Right)

$p_3 = -\Delta y, \; q_3 = y_1 - y_{min}$ (bottom)

$p_4 = \Delta y, \; q = y_{max} - y_1$ (Top)

$u_1 = max(0, r_1, r_3)$

$u_2 = min(1, r_2, r_4)$

(Next অন্য খাতায়)

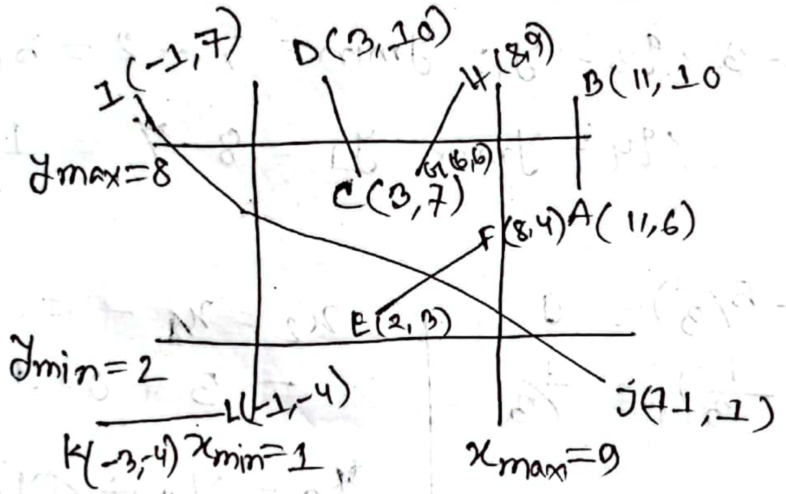
04.03.25

Graphics

Liang Barsky Line Clipping

1. If $P_k = 0 \rightarrow$
   - $\rightarrow$ if $q_k < 0$, the line is completely outside
   - $\rightarrow$ if $q_k \geq 0$ the line is in inside and needs further consideration.

2. $P_k < 0$, the line proceeds outside to inside
3. If $P_k > 0$ " " " inside " outside
4. If $P_k \neq 0$, $u = r_k = q_k / P_k$



Q. Clip those lines using Liang Barsky Clipping.

For line AB:

$P_1 = -\Delta x = 0$, $q_1 = x_1 - x_{min} = 11 - 1 = 10$;
$P_2 = \Delta x = 0$, $q_2 = x_{max} - x_1 = 9 - 11 = -2$
$P_3 = -\Delta y = -4$, $q_3 = y_1 - y_{min} = 6 - 2 = 4$
$P_4 = \Delta y = 4$, $q_4 = y_{max} - y_1 = 8 - 6 = 2$

Since, $P_2 = 0$ and $q_2 = -2 < 0$; completely outside.

For line CD:

$P_1 = -\Delta x = 0$, $q_1 = x_1 - x_{min} = 3 - 1 = 2$
$P_2 = \Delta x = 0$ $q_2 = x_{max} - x_1 = 9 - 3 = 6$
$P_3 = -\Delta y = -3$, $q_3 = y_1 - y_{min} = 7 - 2 = 5$, $r_3 = \frac{-5}{3}$
$P_4 = \Delta y = 3$, $q_4 = y_{max} - y_1 = 8 - 7 = 1$, $r_4 = \frac{1}{3}$

$u_1 = \max(0, -5/3) = 0$
$u_2 = \min(1, \frac{1}{3}) = 1/3$

$\therefore x_1 = x_1 + \Delta x . u_1$
$= 3 + 0.0 = 3$

$y_1 = y_1 + \Delta y . u_1$
$= 7 + (-3.0)$
$= 7$

$\therefore C(3,7)$

$x_2 = x_1 + \Delta x . u_2$
$= 3 + 0 \times 1/3 = 3$

$y_2 = y_1 + \Delta y . u_2$
$= 7 + (3) . 1/3$
$= 7 + 1 = 8$

$D'(3,8)$

For Line GH:

$P_1 = -2, \quad q_1 = x_1 - x_{min} = 6 - 1 = 5, \quad r_1 = -5/2$

$P_2 = 2, \quad q_2 = x_{max} - x_1 = 9 - 6 = 3 = 3/2$

$P_3 = -3, \quad q_3 = y_1 - y_{min} = 6 - 2 = 4 = -4/3$

$P_4 = 3, \quad q_4 = y_{max} - y_1 = 8 - 6 = 2 = 2/3$

$u_1 = \max(0, -5/2, -4/3) = 0$

$u_2 = \min(1, 3/2, 2/3 = 2/3$

$\therefore G \equiv (6,6)$

$x_2 = x_1 + \Delta x . u_2$
$= 6 + 2 . 2/3$
$= 6 + 4/3$
$= 7.33$

$y_2 = y_1 + \Delta y . u_2$
$= 6 + 3 \times 2/3$
$= 8$

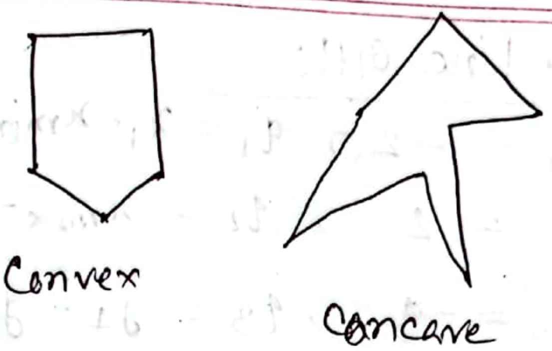
$\therefore H' (7.33, 8)$

"Polygon clipping"

05.03.2024

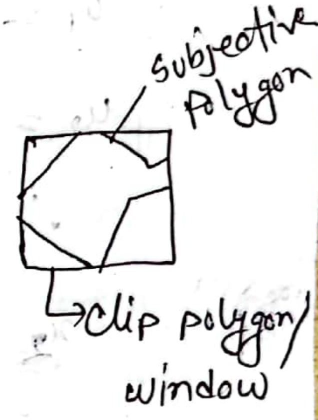
Sides ≥ 4

- → Convex
- → Concave

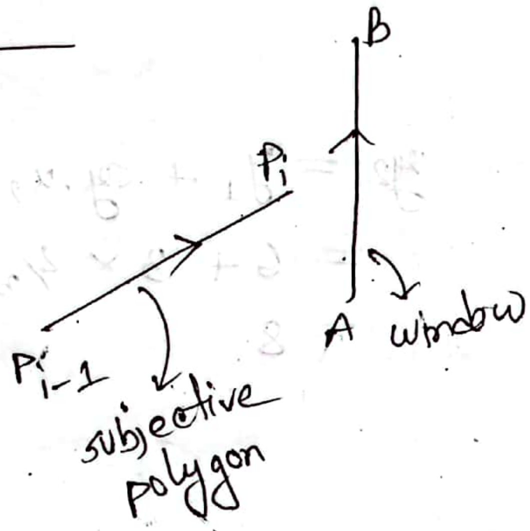


দুই জায়গায় ছেদ করলে convex
" " বেশি " " concave

Sutherland Hodgeman Polygon Clipping Algo :



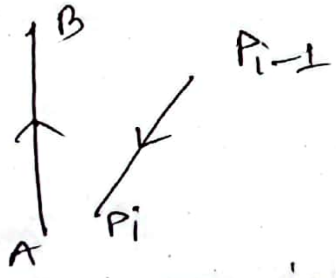
Rules : 1



If both P_{i-1} & P_i are in left of $\vec{AB}$

↳ output = P_i , completely inside

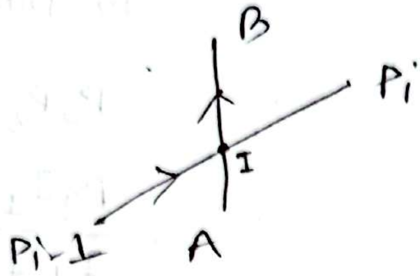
Rules : 2



If both P_i & P_{i-1} are in right of AB.

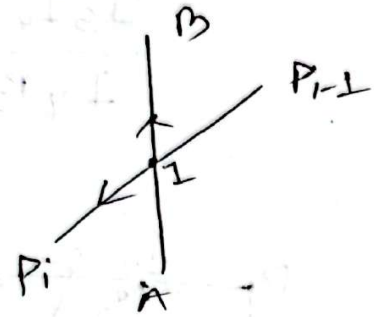
output = nothing, completely outside.

Rules: 3



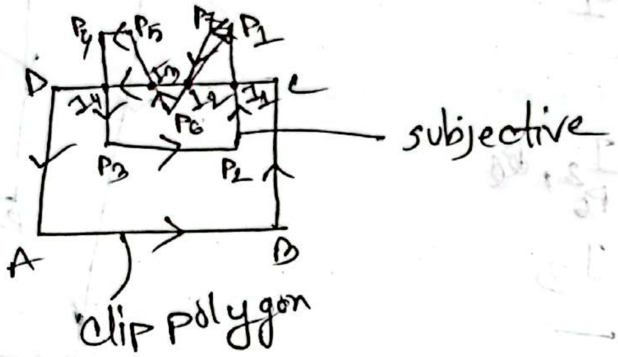
If P_{i-1} is in left & P_i is in right of AB, output = I, inside to outside.

Rules: 4



If P_i is in left and P_{i-1} is in Right, of AB ⇒ output = I, P_i outside to inside

Example:



(1) For $\overline{AB}$ boundary Line:

$\overline{P_3P_2}$ ⟶ output = P_2

$\overline{P_2P_1}$ ⟶ P_1

$\overline{P_1P_7}$ ⟶ P_7

$\overline{P_7P_6}$ ⟶ P_6

$\overline{P_6P_5}$ ⟶ P_5

$\overline{P_5P_4}$ ⟶ P_4

$\overline{P_4P_3}$ ⟶ P_3

(ii) For $\overline{BC}$ boundary line :

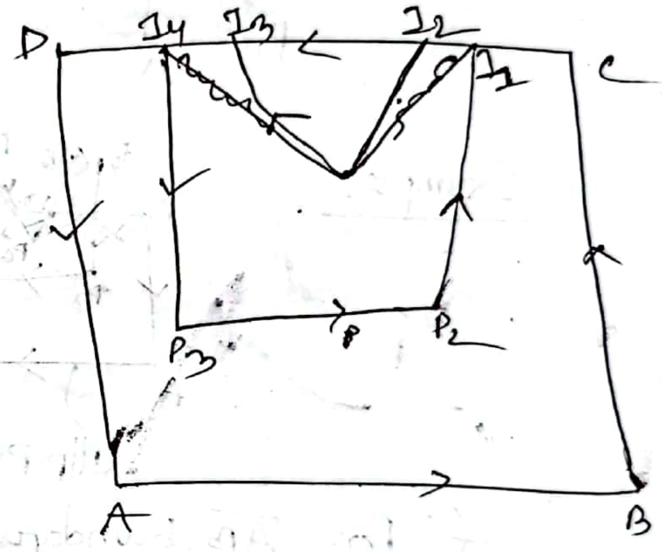
	Output
$\overline{P_3P_2} \rightarrow$	P_2
$\overline{P_2P_1} \rightarrow$	P_1
$\overline{P_1P_7} \rightarrow$	P_7
$\overline{P_7P_6} \rightarrow$	P_6
$\overline{P_6P_5} \rightarrow$	P_5
$\overline{P_5P_4} \rightarrow$	P_4
$\overline{P_4P_3} \rightarrow$	P_3

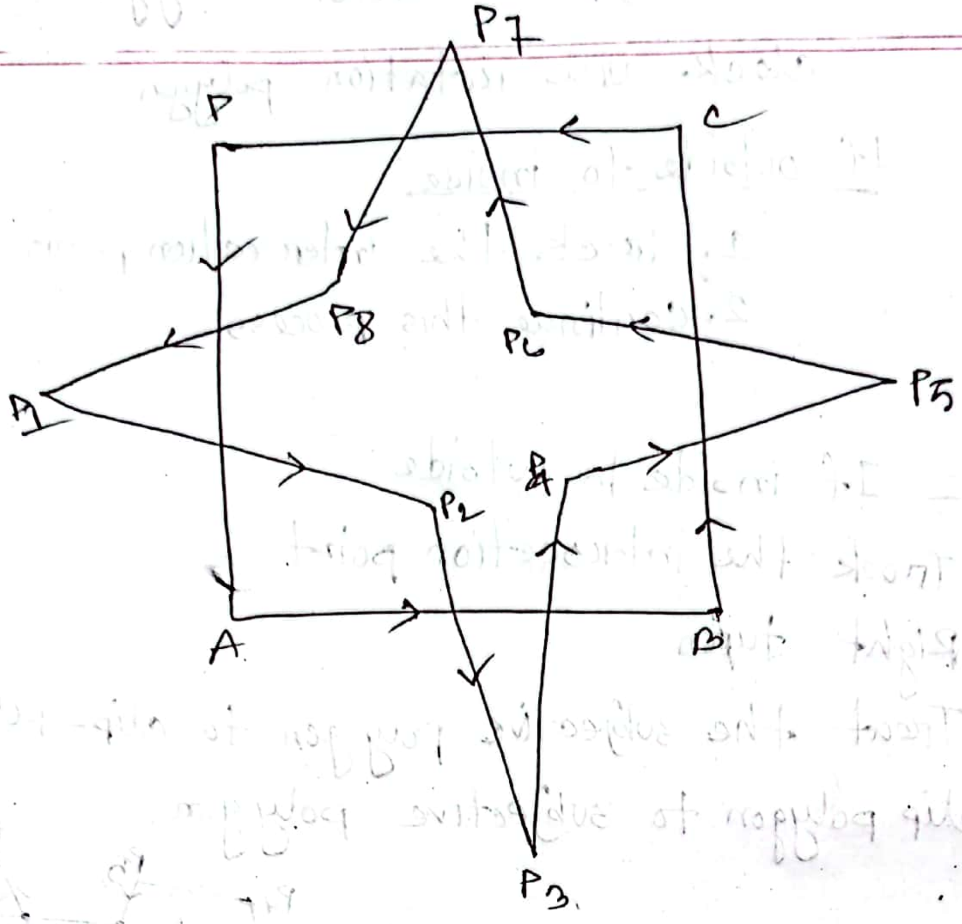
(iv) For $\overline{DA}$ boundary:

	output
$\overline{P_3P_2} \rightarrow$	P_2
$\overline{P_2I_1} \rightarrow$	I_1
$\overline{I_1I_2} \rightarrow$	I_2
$\overline{I_2P_6} \rightarrow$	P_6
$\overline{P_6I_3} \rightarrow$	I_3
$\overline{I_3I_4} \rightarrow$	I_4
$\overline{I_4P_3} \rightarrow$	P_3

(iii) For $\overline{CD}$ boundary line :

	Output
$\overline{P_3P_2} \rightarrow$	P_2
$\overline{P_2P_1} \rightarrow$	I_1
$\overline{P_1P_7} \rightarrow$	—
$\overline{P_7P_6} \rightarrow$	I_2, P_6
$\overline{P_6P_5} \rightarrow$	I_3
$\overline{P_5P_4} \rightarrow$	—
$\overline{P_4P_3} \rightarrow$	I_4 P_3





P7
P
C
P8
P6
P1
P5
P4
P2
A
B
P3

"Wilerc Autherton Polygon Clipping Algo"

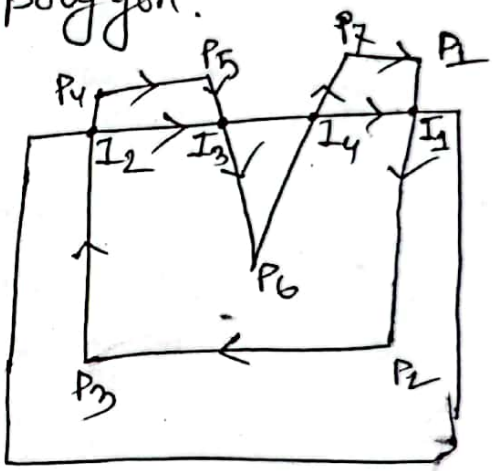
clock-wise rotation polygon

Rules: 1 If outside to inside

1. Track the intersection point
2. Continue this process

Rules: 2 If inside to outside:

1. Track the intersection point
2. Right turn
3. Treat the subjective polygon to clip-polygon and clip polygon to subjective polygon.



4.

Rules: 3 Otherwise leave.

For $\overline{P_1P_2}$: outside to inside and intersection point I_1.

For $\overline{P_2P_3}$: completely inside, leave

For $\overline{P_3P_4}$: inside to outside intersection point I_2

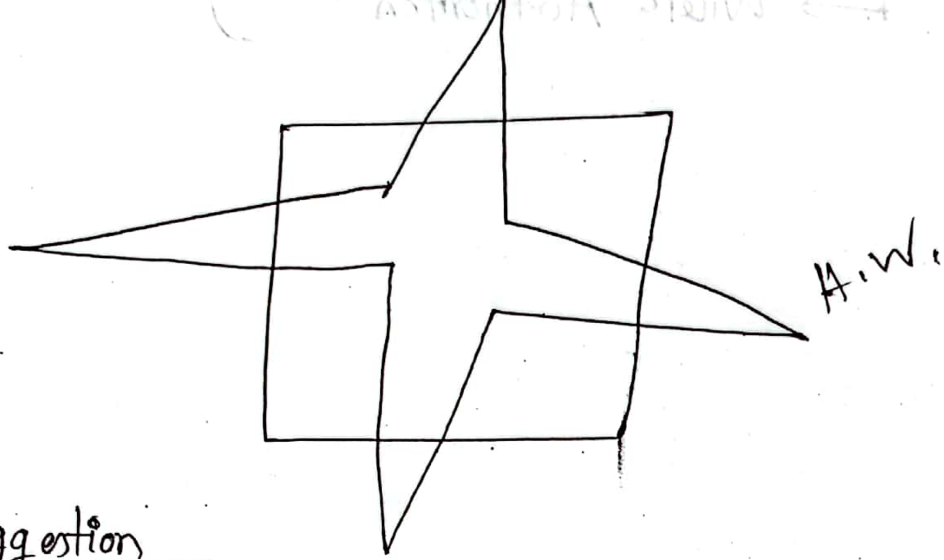
for I_2I_3 : inside to outside intersection point I_3

for $\overline{I_3P_6}$: completely inside leave

For $\overline{P_6 P_7}$: inside to outside.
intersection point I_4

For $\overline{I_4 I_1}$: completely inside
leave

Final Output আঁকাইতে হবে।



Suggestion

1. Introduction (3 slides) ⟶ pixel, vector vs raster, CRT, Image Resolution, Definition of computer graphics, image types, ~~pixel ral~~ aspect ratio, bitmapped
↳ 10 marks

10 marks
2. Line Drawing ⟶
   - ⟶ DDA (Math)
   - ⟶ Bressenham (Derivation + Math) *** :
   - ⟶ Mid point (" ")

3. Circle Drawing ⟶
   - ⟶ Bressenham (Math)
   - ⟶ Mid Point (Math)

4

4. Line clipping :

- Cohen Sutherland

5. Polygon clipping :

- Sutherland Hodgeman
- Wiler Authorton

10 marks