Name :- Udoy Saha

ID :- 23341134

Section :- 11

Ans to the ques no :- 1

[a]

I would choose MPL as an ideal algorithm to draw a line over DDA.

The reasons are stated below :-

1. DDA is slope dependent. MPL can be made slope independent

2. DDA does not ensure endpoint accuracy, while MPL ensures it.

3. The lines are often broken in DDA. However, MPL ensures a somewhat smoother line.

~~4. MPL can~~ 4. Floating points are avoided in MPL.

Because of these reasons, I would prefer DDA.

$\boxed{b}$

Given ~~b~~ line, $y = -2.5x + 10$

At y-intersection, $x = 0$

$\therefore y = 10$

At x-intersection, $y = 0$

$\therefore 0 = -2.5x + 10$

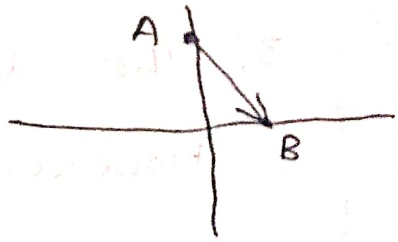
$\Rightarrow x = \frac{10}{2.5}$

$\therefore x = 4$

So, the points are $A(0,10)$ and $B(4,0)$

Here, $dx = 4 - 0 = 4$

$dy = 0 - 10 = -10$

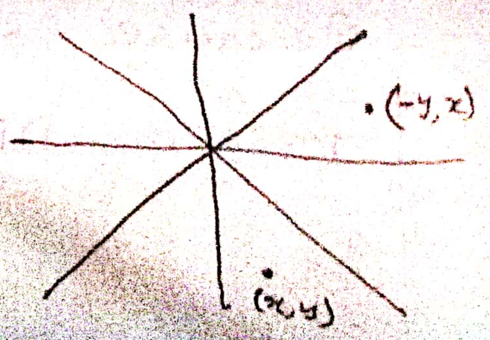


As, dx is positive and dy is negative,
And $|dy| > |dx|$, so it is in Zone-6.

So, A and B ~~converted~~ translated into Zone-0,

$A(-10, 0)$

$B(0, 4)$

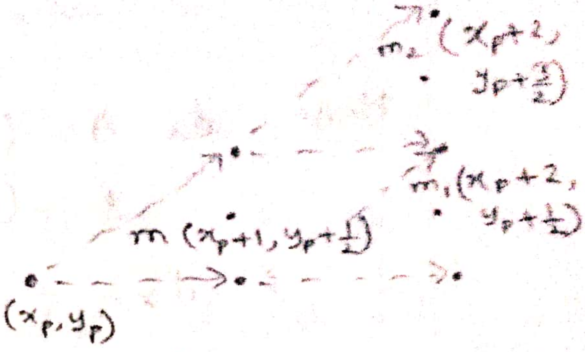


for zone-0,

$$f(m) = A(x_p+1)+B(y_p+\tfrac{1}{2})+C$$

$$= Ax_p + A + By_p + \frac{B}{2} + C$$

$$= (Ax_p + By_p + C) + (A + \frac{B}{2})$$



$$\therefore d_m = A + \frac{B}{2} = dy - \frac{dx}{2}$$

$$d_E = f(m_1) - f(m) \Rightarrow A(x_p+2) + B(y_p+\tfrac{1}{2}) + C$$

$$(-)\ A(x_p+1) + B(y_p+\tfrac{1}{2}) + C$$

$$Ax_p + 2A - Ax_p - A$$

$$= A$$

$$\therefore d_E = A = dy$$

$$\text{And, } d_{NE} = f(m_2) - f(m) \Rightarrow A(x_p+2) + B(y_p+\tfrac{3}{2}) + C$$

$$A(x_p+1) + B(y_p+\tfrac{1}{2}) + C$$

$$2A - A + \frac{3}{2}B - \frac{1}{2}B$$

$$= A + B$$

$$\therefore d_{NE} = A + B = dy - dx$$

(c)

Here, ~~dx~~ $A = (-10, 0)$

$B = (0, 4)$

$\therefore \; dx = 0 - (-10) = 10$

$dy = 4 - 0 = 4$

So, $d_{init} = d_m = 2dy - dx$ [multiplying 2 to avoid fraction]

$= 2 \times 4 - 10$

$= -2$

$d_E = 2\,dy$

$= 2 \times 4$

$= 8$

$d_{NE} = 2(dy - dx)$

$= 2(4 - 10)$

$= -12$

If, $d < 0$ then, d_E

$d \geq 0$ then, d_{NE}

Pixel	x (Zone 0)	y (Zone 0)	d	x_{real}	y_{real}	d_E/d_{NE}
0	-10	0	-2	0	10	d_E
1	-9	0	$-2+8=6$	0	9	d_{NE}
2	-8	1	$6+(-12)=-6$	1	8	d_E
3	-7	1	$-6+8=2$	1	7	d_{NE}
4	-6	2	$2+(-12)=-10$	2	6	d_E
5	-5	2	$-10+8=-2$	2	5	d_E

Ans to the ques no:- 2

[a]

Starting point $(0, P)$

East pixel 10 times makes the point $(0+10, P)$

$= (10, P)$

South-East pixel chosen 6 times. That makes the point $(10+6, P-6)$

$= (16, P-6)$

Ans:- $(16, P-6)$

$\boxed{b}$

If d_{init} is $1.25 - r$, then it will cause issue. Because, we try to eliminate floating point calculations in midpoint algorithm.

We can solve this issue in 2 ways. First, we may take d_{init} as $(1 - r)$ only instead of $(1.25 - r)$. Because, we are only concerned with the sign of d, not the amplitude. In both cases, d_{init} will return non-zero values if r is 0 or 1. In any other case, d will always be ~~be~~ negative whether it is $(1.25 - r)$ or $(1 - r)$.

Another approach is, we see that,

$$1.25 - r = \frac{5}{4} - r .$$

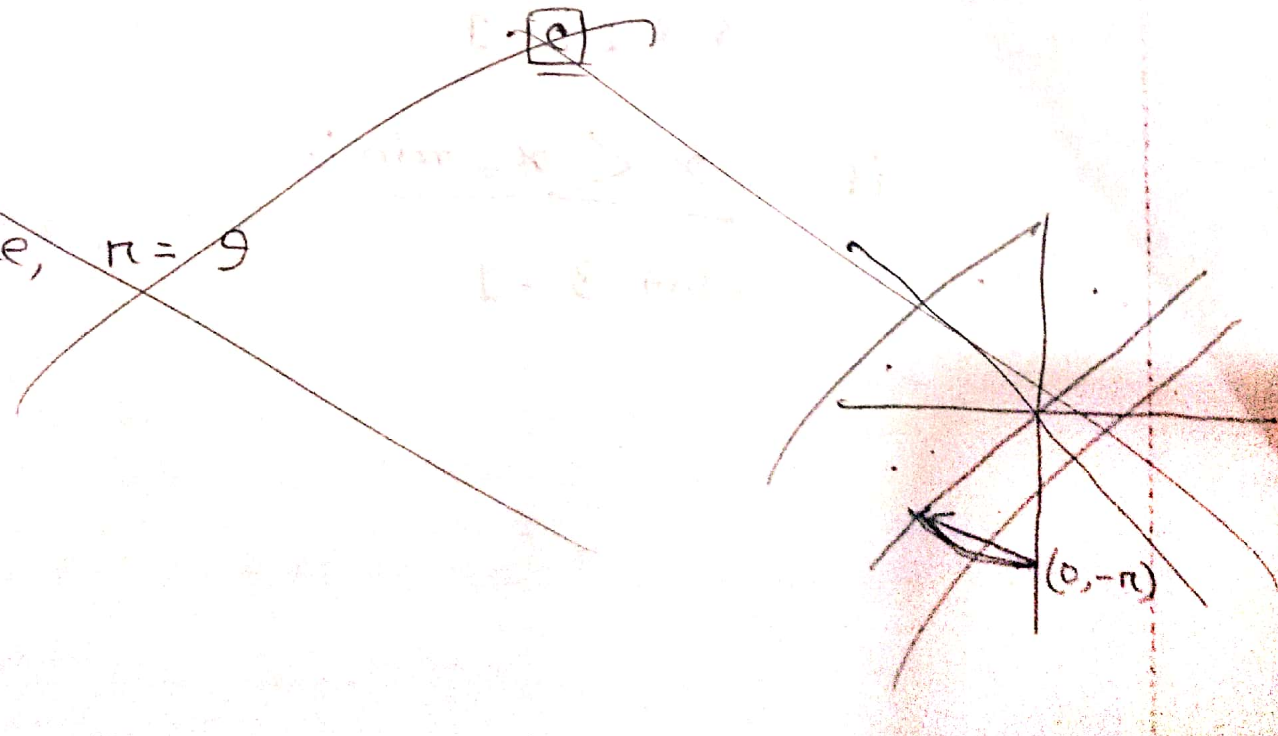
If we multiply 4 with it, then,

$$d_{init} = 5 - 4r .$$

So, fractional values are eliminated.

However we have to multiply 4 with other 2 decision parameters as well.

Here, $r = 9$



Ans to the ques no:-3

$\boxed{a}$

```
def calculate_outcode (x, y):
    bit 0 = bit 1 = bit 2 = bit 3 = 0
    if y > y_max :
        bit 0 = 1
    if y < y_min :
        bit 1 = 1
    if x > x_max :
        bit 2 = 1
    if x < x_min :
        bit 3 = 1
```

Ans

$\boxed{b}$

At most 6 clippings of a line is done while ~~k~~ clipping 3D line.

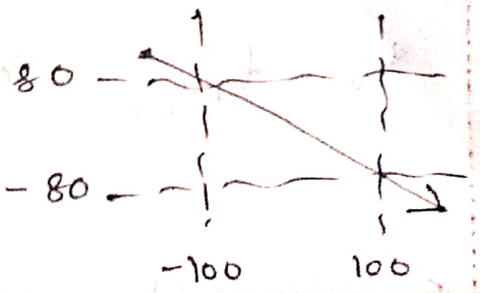
The intersection points are :-

Near, Far, Top, Bottom, Right, Left.

$\boxed{c}$

Let,

$t_{E_{max}} = 0$



$t_{L_{min}} = 1$

$(P_1 - P_0) = (150-(-160))\hat{i} + (-88-90)\hat{j}$

$= 310\hat{i} - 178\hat{j}$

For left edge:-

$N = -\hat{i}$

$(P_1 - P_0) \cdot N = -310$, so its not parallel.

$$t = -\frac{(P_0 - P_E) \cdot N}{(P_1 - P_0) \cdot N}$$

$$= -\frac{\{(-160 - (-100))\hat{i} + (90 - 80)\hat{j}\} \cdot N}{-310}$$

$= 0 \cdot 19$

As, $(P_1 - P_0) \cdot N < 0$, its an entering edge.

So, $t_{E\,max} = 0 \cdot 19$

For right edge:-

$N = +\hat{i}$

$(P_1 - P_0) \cdot N = 310$, so its not parallel

$$t = -\frac{\{(-160-100)\hat{i} + (90-(-80))\hat{j}\} \cdot N}{310}$$

$= 0.84$

Its an leaving edge, so

$t_{L\,min} = 0.84$

<u>For Top edge:-</u>

$N = +\hat{j}$

$(P_1 - P_0) \cdot N = -178$

$$t = -\frac{\{(-160-100)\hat{i} + (90-80)\hat{j}\} \cdot N}{-178}$$

$= 0.056$

Its an entering edge but $t_{E\,max}$ is already higher than it.

CamScanner

For ~~leaving~~ bottom edge:-

$N = -\hat{j}$

$(P_1 - P_0) \cdot N = 178$

$$\therefore t = -\frac{\{(160-(100))\hat{i} + (90-(-80))\hat{j}\} \cdot N}{178}$$

$= 0.95$

Its a leaving edge, but

$t_{L_{min}}$ is already lower than this.

So,

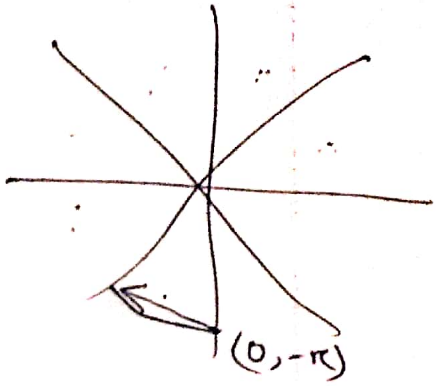
Finally, $t_{E_{max}} = 0.19$

$t_{L_{min}} = 0.84$

—o—x—o—

Ans to the ques no:- 2

$\boxed{c}$



$r = 9$

$\therefore (x_p, y_p) = (0, -9)$

so,

$$d_{init} = -8x_p + 4y_p + 5$$

$$d_W = -8x_p + 12$$

$$d_{NW} = -8x_p + 8y_p + 20$$

$$d_{init} = -8 \times 0 + 4 \times (-9) + 5 = -31$$

$$\cancel{d_W = -8 \times 0 + 12 = 12}$$

$$\cancel{d_{NW} = -8 \times 0 + 8 \times -9 + 20}$$