Name - Shamir Roy

ID - 20241015

Course - CSE923

Sec → 08

Midterm Exam.

Question II

a) Between DDA and Midpoint line algorithm I would choose Midpoint line algorithm as the ideal one as it has the following advantages over DDA:

1. No rounding calculations
2. More efficient and faster than DDA
3. No floating point calculations.

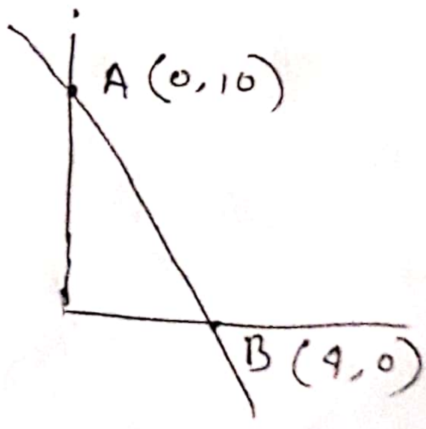
This is why midpoint line algorithm is faster than DDA.

b)

$$y = -2.5x + 10$$

$$\Rightarrow 2.5x + y = 10$$

$$\Rightarrow \frac{x}{4} + \frac{y}{10} = 1.$$



$\therefore$ A (0,10) and B (4,0).

This line falls into zone 5.

Here

$dx = 4$, $dy = -10$

$d_{start} = 2dy - dx = -20 - 4 = -24$

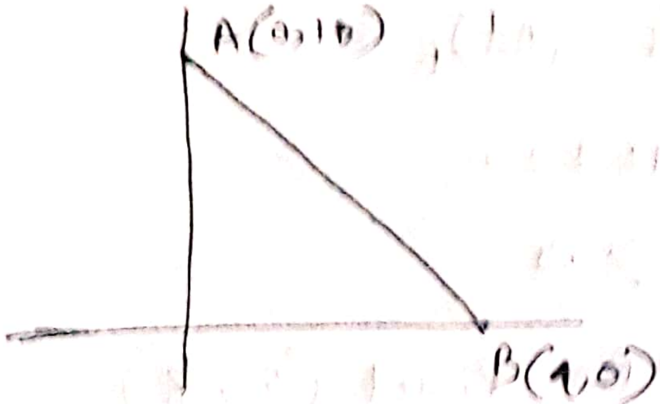
Scanned with CamScanner

$(\Delta d)_E = 2dy = 2x - 10 = -20$

$(\Delta d)_{NE} = 2(dy - dx)$

$= 2(-10-4) = -28.$

c)



$d_{start} = -24 < 0$. A(0,10)

$\therefore$ select E. Pixel $(x+1, y) = (1, 10)$

$\therefore d_1 = d_{start} + (\Delta d)_E$

$= -24 + 20 = -4 < 0$

$\therefore$ select E. Pixel $(x+1, y) = (2, 10)$

$\therefore d_2 = d_1 - (\Delta d)_E$

$= -4 + 20 = 16 > 0$

select NE. Pixel $\cancel{(3, 10)}$ $(3, 9)$

$\therefore$ ~~$d_3 = d_2 + (\Delta d)_E$~~

$d_3 = d_2 + (\Delta d)_{NE}$

$= 16 - 28 = -12 < 0$

select E Pixel $(4, 9)$

$$d_4 = d_3 - (\Delta d)_E$$
$$= -12 + 20$$
$$= 8 > 0$$

Select NE. Pixel (5, 8).

$$d_5 = 8 - 28 = -20 < 0$$

Select E. Pixel (6, 8)

$$d_6 = -20 + 20 = 0$$

Select E, Pixel (7, 8)

$$d_7 = 0 - 20 = -20 < 0$$

∴ Select E, Pixel (8, 8).

	Point	d	Pixel	Zone 0 Pixel	
(0,10)	E	-24	(1,10)		
(1,10)	E	-4	(2,10)		
	NE	16	(3,9)		
	E	-12	(4,9)		
	NE	8	(5,8)		
	E	-20	(6,8)		
	E	0	(7,8)		

	E	8_10	(3,8)

Question 2

a) starting point of a circle (0, P)

zone - 1.

E $\rightarrow$ 10

SE $\rightarrow$ 6

b) In Midpoint circle drawing algorithm the value of dinit is 1.25 − r. ~~but we use it as 1 − r. It will not cause any issues~~. It will cause one problem which is floating point calculations which will make the algorithm slower. To resolve it we use it as 1 − r. Thus we don't have to do any floating point calculations, making it faster and more efficient.

c)

zone − 5

$r = 9$ origin $(-4, 13)$

$h = 1 - 9 = 8 > 0$ SE. ~~(0, 9)~~ $(0, -9)$

$h_1 = h + 2(x - y) + 5$

$= 0 + 2 \times 9 + 5 = 23 > 0$ SE. ~~(1, −8)~~ $(1, -10)$

$h_2 = h_1 + 2(x - y) + 5$ $(2, -11)$

$= 23 + 2(1 + 10) + 5 = 50 > 0$ SE ~~(2, −11)~~

$h_3 = 58 + 2(2+11) + 5 = 81 > 0$ SE $\cancel{(3, -13)}$ $(3, -12)$

$h_4 = 81 + 2(3+12) + 5 = 116 > 0$ SE $\cancel{(4, -13)}$ $\cancel{SE(4, -13)}$ $SE(4, -13)$

$h_5 = 116 + 2(4+13) + 5 = 155 > 0$ SE $(5, -14)$

Point	h	Pixel Point	Origin with
(0, -9)	8	SE	(-4, 4)
(1, -10)	23	SE	(-3, 3)
(2, -11)	58	SE	(-2, 2)
(3, -12)	81	SE	(-1, 1)
(4, -13)	116	SE	(0, 0)
(5, -14)	155	SE	(1, -1)

Scanned with CamScanner

3.

a)

```
def calculate_outcode(x, y):
    bit0 = bit1 = bit2 = bit3 = 0
    if (x<xmin):
        bit0 = 1
    if (x>xmax):
        bit1 = 1
    if (y<ymin):
        bit2 = 1
    if (y>ymax):
        bit3 = 1
```

b) 6 possible clippings can be possible for ~~the~~ Cohen-Sutherland line algorithm using 3D line.

The name of the intersection points are:
(l, 0, 0), (-l, 0, 0), (0, l, 0), (0, -l, 0)
(0, 0, l), (0, 0, -l).

Scanned with CamScanner

e) $x_{min} = -100$, $x_{max} = 100$
$y_{min} = -80$, $y_{max} = 80$
$(-100, -80) \rightarrow (100, 80)$

$P(x_0, y_0) = (-160, 90)$ $(x_1, y_1) = (150, -88)$

$D = (x_1 - x_0, y_1 - y_0) = (310, -178)$

$t_{left} = \dfrac{-(x_0 - x_{min})}{(x_1 - x_0)}$ $t_{right} = \dfrac{-(x_0 - x_{max})}{(x_1 - x_0)}$

$t_{top} = \dfrac{-(y_0 - y_{max})}{(y_1 - y_0)}$ $t_{bottom} = \dfrac{-(y_0 - y_{min})}{(y_1 - y_0)}$

Initially, $t_E = 0$, $t_L = 1$

Boundary	N_i	N_iD	t	PE/PL	t_E	t_L
left	$(-1,0)$	-310	$\frac{6}{31}$		~~$\frac{6}{31}$~~ 0	$\frac{6}{31}$
right	$(1,0)$	310	$\frac{26}{31}$		0	$\frac{6}{31}$
Bottom	$(0,-1)$	~~88~~ 178	$\frac{5}{89}$		$\frac{6}{31}$	$\frac{6}{31}$
Top	$(0,1)$	~~[illegible]~~ -178	$\frac{85}{89}$		$\frac{6}{31}$	$\frac{6}{31}$