GOODLUCK | Page No. | Date

Assignment No. 3

1) Write and explain bresenham's generalized line generation algorithm for all quadrant.

→ The line end points are (x_1, y_1) and (x_2, y_2) assumed not equal.
Integer is the integer function.
$x, y, \Delta x, \Delta y$ are assumed integer; e is real initialize variables.

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

initialize e to compensate for a non-zero intercept

$$e = m - \frac{1}{2}$$

```
begin the main loop
for  i=1 to Δx
  setpixel(x,y)
    while (e>0)
        y = y+1
        e = e-1
    end while
    x = x+1
    e = e+m
    next i
  finish.
```

2) Write and explain bresenham's integer algorithm for all quadrants. Apply the algorithm to find out pix

values for line L from (0,0) to (-8,-4) find out value of error term for every pixel $\Delta x, \Delta y$.

→ The line end points are (x_1, y_1) & (x_2, y_2). Assume not equal.

All variables are assumed integer.

The sign functions returns -1, 0, 1 as its argument is less than 0, = 0, or > 0.

Initialize variables

$x = x_1$

$y = y_1$

$\Delta x = abs(x_2 - x_1)$

$\Delta y = abs(y_2 - y_1)$

$s_1 = sign(x_2 - x_1)$

$s_2 = sign(y_2 - y_1)$

Interchange Δx & Δy depending on the slope of line

```
if Δy > Δx then
    temp = Δx
    Δx = Δy
    Δy = temp
    Interchange = 1
else
    Interchange = 0
end if
```

Initialize the error term to compensate for a non-zero intercept

$\bar{e} = 2 * \Delta y - \Delta x$

main loop

```
for i = 1 to Δx
    setpixel (x, y)
    while (ē > 0)
```

if interchange $e = 1$ then
$x = x + s_1$
else
$y = y + s_2$
end if
$\bar{e} = \bar{e} - 2 * \Delta x$
end while
if interchange $= 1$ then
$y = y + s_2$
else
$x = x + s_1$
end if
$\bar{e} = \bar{e} + 2 * \Delta y$
end i
finish.

$x_1 = 0$, $y_1 = 0$
$x_2 = -8$, $y_2 = -4$
$\Delta x = x_2 - x_1 = -8$
$\Delta y = y_2 - y_1 = 4$
$s_1 = -1$
$s_2 = -1$

i	setpixel	$\bar{e}$	x	y
		0	0	0
1	(0,0)	8	-1	0
2	(-1,0)	-8	-2	-1
	(-2,-1)	8	-3	-1
	(-3,-1)	8	-4	-2

3) Write down Bresenham's incremental circle algorithm for first quadrant. Apply the algorithm to find out pixel values for the first quadrant of circle where circle radius is 8. Find out values of Δ_i, δ and δ'.

→ All variables assumed integer
initialize the variables

$x_i = 0$

$y_i = R$

$\Delta_i = 2(1-R)$

Limit = 0

while $y_i \geq$ limit

call set pixel (x_i, y_i)

determine if case 1 or 2, 4 or 5 or 3

if $\Delta_i < 0$ then

$\delta = 2\Delta_i + 2y_i - 1$

determine whether case 1 or 2

if $\delta \leq 0$

call mh(x_i, y_i, Δ_i)

else

call md(x_i, y_i, Δ_i)

end if

else if $\Delta_i > 0$ then

$\delta' = 2\Delta_i - 2x_i - 1$

determine whether case 4 or 5

if $\delta' \leq 0$ then

call md(x_i, y_i, Δ_i)

else

call mv(x_i, y_i, Δ_i)

end if.

else if $\Delta i = 0$ then
call md $(x_i, y_i, \Delta i)$
end if
end while
finish

move horizontally
subroutine mh $(x_i, y_i, \Delta i)$
$x_i = x_i + 1$
$\Delta i = \Delta i + 2x_i + 1$
end sub
move diagonally
subroutine md $(x_i, y_i, \Delta i)$
$x_i = x_i + 1$
$y_i = y_i - 1$
$\Delta i = \Delta i + 2x_i - 2y_i + 2$
end sub
move vertically
subroutine mv $(x_i, y_i, \Delta i)$
$y_i = y_i - 1$
$\Delta i = \Delta i - 2y_i + 1$
end sub

$x_i = 0$

$y_i = r = 8$
$\Delta i = 2(1 - r)$
$= 2(1 - 8)$
$= -14$

for $\Delta i = -14$
$\delta = 2\Delta i + 2y_i - 1$
$= 2(-14) + 2(8) - 1$
$= -13$

$\delta = 2\Delta i - 2x_i - 1$
$= 2(-14) - 2(0) - 1$
$= -28 - 1$
$= -29$

set pixel	Δi	δ	δ'	x	y
(0,8)	-14	-13		0	8
	-3			1	8
(1,8)	-6	-7		2	8
(2,8)	-12	3		3	7
(3,7)	-3	-11		4	7
(4,7)	-3	7		5	6
(5,6)	1	5		6	5
(6,5)	3		-11	7	4
(7,4)	4		-3	7	3
(7,3)	15		-7	8	2
(8,2)	17		19	8	1
(8,1)	18		17	8	0

4) Explain seed fill algorithm for 4 connected boundary defined regions.

→ 1) seed (x,y) is the seed pixel.
2) Push is a function for placing a pixel on stack.
3) Pop is a function for removing a pixel from stack.

Pixel (x,y) = seed pixel
initilize stack
Push pixel (x,y)
if Pixel (x,y) <> new value then
    Pixel (x,y) = new value.
end if

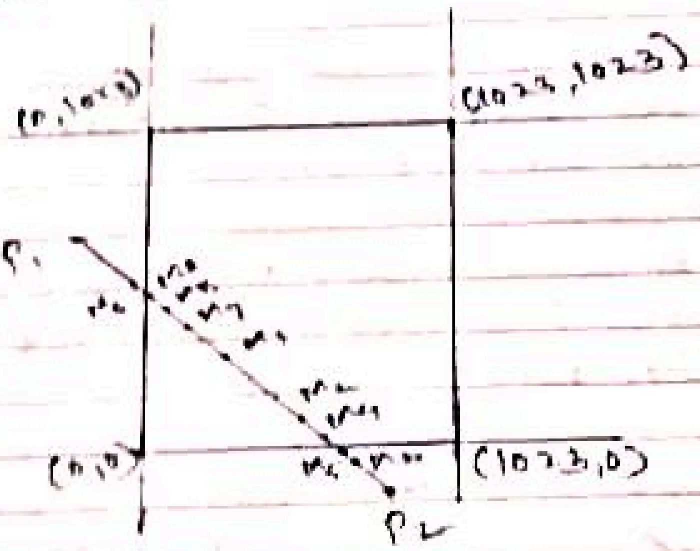
examine the surrounding pixels to see if they should be placed onto the stack

if (pixel (x+1, y) <> new value and
pixel (x+1, y) <> Boundary value) then
push pixel (x, y+1)
end if
if (pixel (x, y+1) <> new value and
pixel (x, y+1) <> Boundary value) then
push pixel (x, y+1)
end if
if (pixel (x-1, y) <> new value and
pixel (x-1, y) <> Boundary value) then
push pixel (x-1, y)
end if
if (pixel (x, y-1) <> new value and
pixel (x, y-1) <> Boundary value) then
push pixel (x, y-1)
end if
end while
finish

Simple seed fill algorithms for a boundary defined region can be developed using stack. Stack is an array, into which values are sequentially placed or from which they are sequentially removed. As new value pushed onto the stack, all previously stored values are pushed down one level. As values are removed or popped from the stack, previously stored values pop up one level. Such a stack is referred to as a first in last out (FILO) or push down stack.

Consider the window in the screen co-ordinates to have left, right, bottom and top edges of 0, 1023, 0, 1023 resp. Apply midpoint subdivisions algorithm on line (

with end points $P_1(-307, 631)$ and $P_2(820, -136)$ to find all visible portion of line. find all the 4 bit code for P_1 and P_2



$P_1(-307, 631)$, $P_2(820, -136)$

midpoint of P_1P_2 is M_1

$x = \frac{820 - 307}{2} = 256$ $\quad y = \frac{-136 + 631}{2} = 247$

$M_1(256, 247)$

M_1 is inside the clipping window.

consider the segment M_1P_2

midpoint of M_1P_2 is M_2

$x = \frac{820 + 256}{2} = 538$, $\quad y = \frac{-136 + 247}{2} = 55$

$M_2(538, 55)$

M_2 is inside the clipping window

$\therefore$ consider the segment M_2P_2

Midpoint of M_2P_2 is M_3

$x = \frac{820 + 538}{2} = 679$ $\quad y = \frac{-136 + 55}{2} = -40$

$M_3 = (679, -40)$

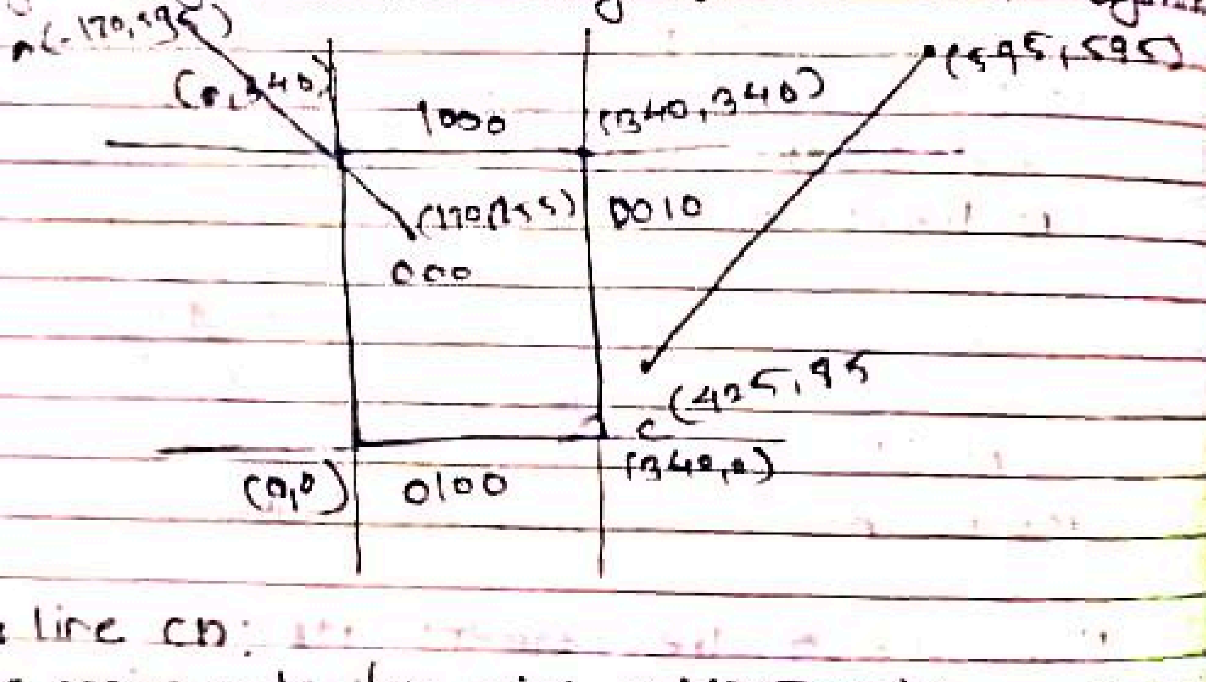
m_6 is inside the clipping window
consider segment m_6m_8
m_9 is the midpoint of m_6m_8

$x = \frac{45-25}{2} = 10$ $\quad y = \frac{391+439}{2} = 415$

$m_9(10, 415)$
M_9 is inside the clipping window.
visible portion of line is m_9m_4
$m_9(10, 415)$ $\quad m_4(108, 7)$

4-bit code for P_1 is 0001 & for P_2 is 0100

1.c. Given the clipping window P(0,0), Q(340,0) & S(0,340) R(340, 340). find out visible portion of life AB[(-170, 595), (170, 255)] & CD[(425, 85), (595, 595)] against given window using cohensutheland algorithm



for line CD:
The region code for point C(425, 85) is 0010 i.e. non zero
The region code for point D(595, 595) is 1010 i.e. non zero.

GOOD LUCK Page No.
Date

$$\begin{array}{r} 0010 \\ \textcircled{T}\ 1010 \\ \hline 0010 \end{array}$$

The Anding result is non 0
∴ The line CD is completely invisible

for line AB.
The region code for point A(-170, 595) is 1001
i.e. non-zero
The region code for point B(170, 255) is 0000

$$\begin{array}{r} 1001 \\ \textcircled{T}\ 0000 \\ \hline 0000 \end{array}$$

The Anding operation result is zero
∴ The line is partially visible

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

$(x_1, y_1) = (-170, 595)$
$(x_2, y_2) = (170, 255)$

$m = \frac{255 - 595}{170 + 170} = \frac{-340}{340} = -1$

$m = -1$

As here the given line AB intersect to the left edge of the window.

$x_L = 0$

$x_1, \quad y = m(x_L - x_1) + y_1$

$y = -1(0 + 170) + 595$

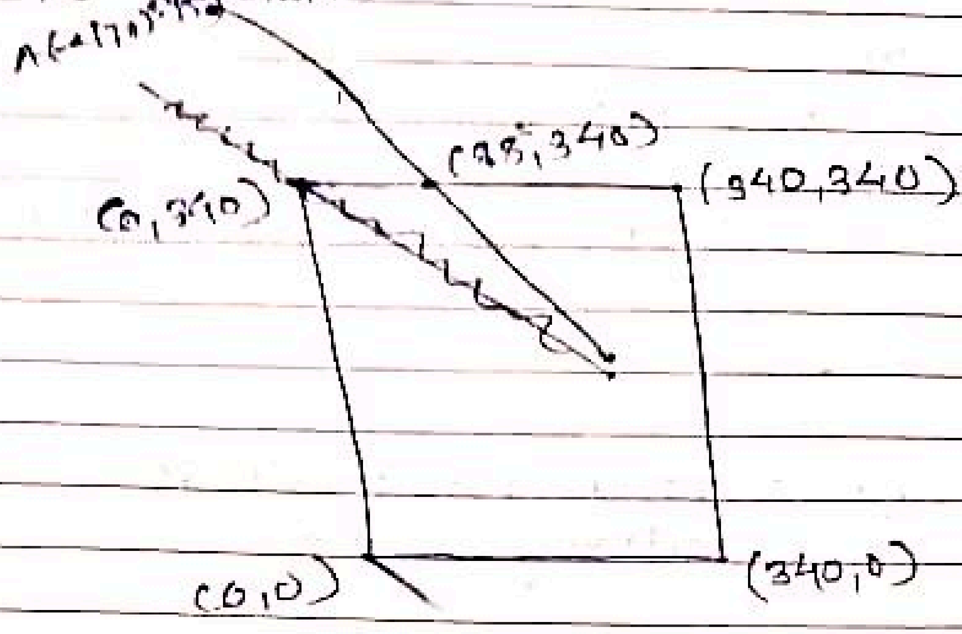
$y = -170 + 595$

$y = 425$

$(x_1, y_1) - (0, 405)$

Let $(0, 425) = x$

∴ The line portion xB [(0,425), (170,255)] is visible.



line seg intersects the top edge of window then top intersection is,

y_T, $x = x_1 + \left(\frac{1}{m}\right)(y_T - y_1)$

$= -170 + (-1)(340 - 595)$

$= -170 + 255$

$x = 85$

$(85, 340)$