DDA (Digital Differential Analyzer) Algorithm

$\Delta y = m \cdot \Delta x$

1. If the line is processed from left to right i.e. left end point is starting point.

   i. $m < 1$, $\Delta x = 1$

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

   ii. $m > 1$, $\Delta y = 1$

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

2. If the line is processed from right to left i.e. right end point is starting point.

   i. $m < 1$, $\Delta x = -1$

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

   ii. $m > 1$, $\Delta y = -1$

   $x_{k+1} = x_k - \frac{1}{m}$

Calculate the intermediate pixel positions of a line with end points (5,15) and (12,10).

Soln;

Given, $(x_1, y_1) = (5,15)$, $(x_2, y_2) = (12,10)$

$dx = x_2 - x_1 = 12 - 5 = 7$

$dy = y_2 - y_1 = 10 - 15 = -5$

$\frac{dy}{dx} = \frac{-5}{7}$

Left to right, $m < 1$

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

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

Initially, (5,15) was plotted.

$x_{k+1} = x_k + 1$	$y_{k+1} = y_k + m$	Point to plot (x, y)
$5 + 1 = 6$	$15 + \left(\frac{-5}{7}\right) = \frac{100}{7} = 14.3$	(6,14)
$6 + 1 = 7$	$\frac{100}{7} + \left(\frac{-5}{7}\right) = \frac{95}{7} = 13.6$	(7,14)
$7 + 1 = 8$	$\frac{95}{7} + \left(\frac{-5}{7}\right) = \frac{90}{7} = 12.9$	(8,13)
$8 + 1 = 9$	$\frac{90}{7} + \left(\frac{-5}{7}\right) = \frac{85}{7} = 12.1$	(9,12)
$9 + 1 = 10$	$\frac{85}{7} + \left(\frac{-5}{7}\right) = \frac{80}{7} = 11.4$	(10,11)
$10 + 1 = 11$	$\frac{80}{7} + \left(\frac{-5}{7}\right) = \frac{75}{7} = 10.71$	(11,11)
$11 + 1 = 12$	$\frac{75}{7} + \left(\frac{-5}{7}\right) = \frac{70}{7} = 10$	(12,10)

DDA Algorithm

- End points (x_1, y_1) and (x_2, y_2)
- $dx = x_2 - x_1$ and $dy = y_2 - y_1$
- if (abs(dx) > abs(dy))

```
        steps = abs(dx);
else
        steps = abs(dy);
```

- $incrx = \frac{dx}{steps}$ and $incry = \frac{dy}{steps}$
- $(x, y) = (x_1, y_1)$ /* first point to plot */
- putpixel (round(x), round(y),1); //1 is color parameter
- for(k=1; k<=steps; k++)

```
{
    x = x + incrx;
    y = y + incry;
    putpixel(round(x), round(y),1);
}
```

Program in C:

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
void main()
{
    int gd= DETECT, gm, x1=100, y1=100, x2=200,
    y2=200, steps, k, dx, dy;
      float incrx, incry, x, y;
    initgraph(&gd, &gm, "C:\\TC\\BGI");

    dx = x2-x1;
    dy = y2-y1;
    steps = abs(dx) > abs(dy) ? abs(dx) : abs(dy);
    incrx = dx/steps;
```

```
incry = dy/steps;
x = x1;
y = y1;
putpixel(x, y, RED);
for(k=1; k<=steps; k++)
{
    x = x+incrx;
    y = y+incry;
    putpixel(x, y, RED);
}
getch();
closegraph();
}
```

Similarly,

$P_{k+1} = 2(\Delta y) x_{k+1} + 2(\Delta y) - 2(\Delta x) y_{k+1} + b(\Delta x) \cdots (3)$

So,

$P_{k+1} - P_k = 2(\Delta y)(x_{k+1} - x_k) - 2(\Delta x)(y_{k+1} - y_k)$

$P_{k+1} = P_k + 2(\Delta y) - 2(\Delta x)(y_{k+1} - y_k) \cdots (4)$

If p_k is $-ve$,

$y_{k+1} = y_k$

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

Else,

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

$P_{k+1} = P_k + 2(\Delta y) - 2(\Delta x)$

For initial P_0, let the point be (x_0, y_0).

From equation (1),

$d_1 - d_2 = 2m(x_0 + 1) + 2c - 2y_0 - 1$

$d_1 - d_2 = 2mx_0 + 2m + 2c - 2(mx_0 + c) - 1$

$= 2mx_0 + 2m + 2c - 2mx_0 - 2c - 1$

$= 2m - 1$

~~$\Delta x(d_1 - d_2) = 2\Delta y - \Delta x$~~

$d_1 - d_2 = 2\frac{\Delta y}{\Delta x} - 1$

$\Delta x(d_1 - d_2) = 2\Delta y - \Delta x$

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

(3)

(4)

Q. (5,15) to (12,10) using Bresenham algorithm.

Solution:

$(x_1, y_1) = (5,15)$, $(x_2, y_2) = (12,10)$

Since $x1 < x2$, $x\ inc = +1$

Since $y1 > y2$, $y\ inc = -1$

$dx = |x_2 - x_1| = |12-5| = 7$

$dy = |y_2 - y_1| = |10-15| = 5$

$m = \frac{dy}{dx} = \frac{5}{7} \Rightarrow m < 1$

$P_0 = 2dy - dx = 2*5-7 = 3$

Initially, (5,15) was plotted

P_k	$x_{k+1} = x_k + x\ inc$	$y_{k+1} = y_k$ or $y_k + y\ inc$	P_{k+1}
3	5+1=6	15-1=14	3+2×5-2×7=-1
-1	6+1=7	14	-1+2×5=9
9	7+1=8	14-1=13	9+2×5-2×7=5
5	8+1=9	13-1=12	5+2×5-2×7=1
1	9+1=10	12-1=11	1+2×5-2×7=-3
-3	10+1=11	11	-3+2×5=7
-7	11+1=12	11-1=10	7+2×5-2×7=3

Hence, (x_{k+1}, y_{k+1}) are the points to be plotted.

Code in C:

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
void main()
{
 int gd= DETECT, gm, x1=100, y1=100, x2=300,
 y2=300;
 int dx, dy, xinc, yinc, steps, p, x, y, k;
```

```
initgraph(&gd, &gm, "C: \\TC\\BGI");
xinc = (x2>x1) ? 1:-1;
yinc = (y2>y1) ? 1:-1;
dx = abs(x2-x1);
dy = abs(y2-y1);

if (dx>dy)
{
    steps = dx;
    p = 2*dy-dx;
    x=x1; y=y1;
    putpixel(x,y,RED);
    for(k=1; k<=steps; k++)
    {
        x = x+xinc;
        if(p<0)
            y=y;
            p=p+2*dy;
        }
        else
        {
            y=y+yinc;
            p=p+2*dy-2*dx;
        }
        putpixel(x,y,RED);
    }
}
else
{
    steps=dy;
    p=2*dx-dy;
    x=x1; y=y1;
    putpixel(x,y,RED);
```

```
for(k = 1; k<= steps; k++)
{
    y = y + yinc;
    if ( p < 0)
    {
        x = x;
        p = p + 2 * dx;
    }
    else
    {
        x = x + xinc;
        p = p + 2 * dx - 2 * dy;
    }
    putpixel(x, y, RED);
}
}
getch();
closegraph();
}
```

Mid-point Circle Drawing Algorithm

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

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

$f_{circle} = x^2 + y^2 - r^2$

$P_k = f(x_k + 1, y_k - \frac{1}{2})$

$P_{k+1} = f(x_{k+1} + 1, y_{k+1} - \frac{1}{2})$

$= f(x_k + 1 + 1, y_{k+1} - \frac{1}{2})$

Now,

$P_{k+1} - P_k = f(x_k + 2, y_{k+1} - \frac{1}{2}) - f(x_k + 1, y_k - \frac{1}{2})$

$= (x_k + 2)^2 + (y_{k+1} - \frac{1}{2})^2 - r^2 - [(x_k + 1)^2 + (y_k - \frac{1}{2})^2 - r^2]$

$= x_k^2 + 4x_k + 4 + y_{k+1}^2 - y_{k+1} + \frac{1}{4} - r^2 - x_k^2 - 2x_k - 1 - y_k^2$

$+ y_k - \frac{1}{4} + r^2$

$= 2x_k + 3 + y_{k+1}^2 - y_k^2 - y_{k+1} + y_k$

$= 2x_k + 2 + 1 + (y_{k+1}^2 - y_k^2) - (y_{k+1} - y_k)$

$= 2(x_k + 1) + 1 + (y_{k+1}^2 - y_k^2) - (y_{k+1} - y_k)$

$P_{k+1} = P_k + 2x_{k+1} + 1 + (y_{k+1}^2 - y_k^2) - (y_{k+1} - y_k)$

If p_k is -ve,

$y_{k+1} = y_k$ (out of circle)

$P_{k+1} = P_k + 2x_{k+1} + 1$

else,

$y_{k+1} = y_k - 1$ (inside circle)

$P_{k+1} = P_k + 2x_{k+1} + 1 + [(y_k - 1)^2 - y_k^2] - (y_k - 1 - y_k)$

$\cancel{P_{k+1} = P_k + 2x_k + 1 - 2y_k + 3}$

$P_{k+1} = P_k + 2x_k + 1 + 1 + [y^2_k - 2y_k + 1 - y^2_k] + 1$

$P_{k+1} = P_k + 2x_k + 1 - 2y_k + 3$

$P_{k+1} = P_k + 2x_k + 1 - 2y_k + 2 + 1$

$P_{k+1} = P_k + 2x_k + 1 - 2(y_k - 1) + 1$

$P_{k+1} = P_k + 2x_k + 1 - 2y_k + 1 + 1$

Now,

$P_k = f(x_k + 1, y_k - \frac{1}{2})$

Initial point to be plotted is $(x_0, y_0) = (0, r)$.

So,

$P_0 = f(0 + 1, r - \frac{1}{2})$

$P_0 = (0+1)^2 + (r - \frac{1}{2})^2 - r^2$

$= 1 + r^2 - r + \frac{1}{4} - r^2$

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

$P_0 \approx 1 - r$

Q. Centre of circle is $(0,0)$ and radius $= 10$. Find the pixels to be plotted in the positive x-axis for the 1st octant.

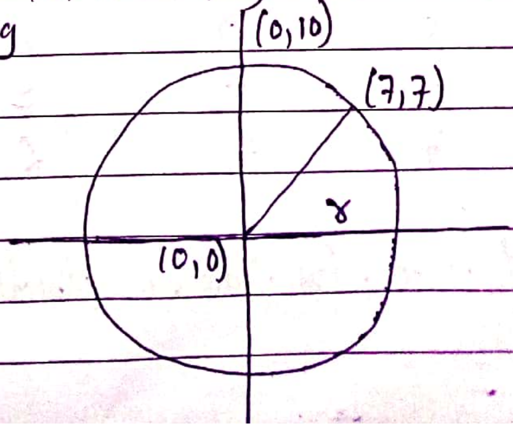
=> Solution;

Centre $= (0,0)$

radius $(r) = 10$

1st point to plot $= (0, r) = (0, 10)$

$P_0 = 1 - r = 1 - 10 = -9$



P_k	$x_{k+1} = x_k + 1$	$y_{k+1} = y_k$ or $y_k - 1$	P_{k+1}
-9	0+1=1	10	-9+2×1+1=-6
-6	1+1=2	10	-6+2×2+1=-1
-1	2+1=3	10	-1+2×3+1=6
6	3+1=4	10-1=9	6+2×4-2×9+1=-3
-3	4+1=5	9	-3+2×5+1=8
8	5+1=6	9-1=8	8+2×6-2×8+1=5
5	6+1=7	8-1=7	5+2×7-2×7+1=6

Hence, (x_{k+1}, y_{k+1}) are the point to be plotted for the 1st octant.

Q. Centre of a circle is (20,30) and radius = 10. Find the pixels to be plotted in the positive x-axis for the 1st octant.

Soln:

Centre = (20,30)

radius (r) = 10

1st point to take = (0, r) = (0,10)

1st point to plot = (20+0, 30+10) = (20,40)

$P_0 = 1 - r = 1 - 10 = -9$

P_k	$x_{k+1} = x_k + 1$	$y_{k+1} = y_k$ or $y_k - 1$	P_{k+1}	Point to plot (x,y)
-9	1	10	-6	(20+1, 30+10) = (21,40)
-6	2	10	-1	(20+2, 30+10) = (22,40)
-1	3	10	6	(20+3, 30+10) = (23,40)
6	4	9	-3	(20+4, 30+9) = (24,39)
-3	5	9	8	(20+5, 30+8) = (25,38)
8	6	8	5	(20+6, 30+8) = (26,38)
5	7	7	6	(20+7, 30+7) = (27,37)

Hence, (x,y) are the pixels to be plotted.

If $P_k < 0$, $y_{k+1} = y_k$

$\therefore P_{k+1} = P_k + 2r_y^2 x_{k+1} + r_y^2$

otherwise $y_{k+1} = y_k - 1$,

$\therefore P_{k+1} = P_k + 2r_y^2 x_{k+1} + r_y^2 - 2r_x^2 y_{k+1}$

Ellipse

1. Draw ellipse at center (0,0) and $r_x = 10$ and $r_y = 8$.

⇒ Solution;

Center = (0,0)

$r_x = 10$, $r_y = 8$

$x^2 \cdot r_y^2 + y^2 \cdot r_x^2 - r_x^2 \cdot r_y^2$

For Region I

We start from $(0, r_y) = (0, 8)$

Initially, (0,8) was plotted

$P_0 = f(0+1, 8-\frac{1}{2}) = (0+1)^2 \times 8^2 + (8-\frac{1}{2})^2 \times 10^2 - 10^2 \times 8^2$

$= -711$

Here,

P_k	$x_{k+1} = x_k + 1$	$y_{k+1} = y_k$ or $y_k - 1$	P_{k+1}	$2r_y^2 x_{k+1}$	$2r_x^2 y_{k+1}$
-711	0+1=1	8	-519	128	1600
-519	1+1=2	8	-199	256	1600
-199	3	8	249	384	1600
249	4	7	-575	512	1400
-575	5	7	129	640	1400
129	6	6	-239	768	1200
-239	7	6	721	896	1200
721	8	5	809	1024	1000

We move out of Region I since $2r_y^2 x_{k+1} > 2r_x^2 y_{k+1}$.

For region II,

First point to take/plot is (8,5) since it is the last point of region I.

Initially;

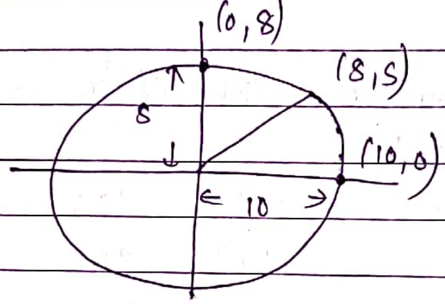
$P_0 = f(8+\frac{1}{2}, 5-1)$

$= (8+\frac{1}{2})^2 \times 8^2 + (5-1)^2 \times 10^2 - 10^2 \times 8^2$

$= -176$

P_k	$x_{k+1}=x_k$ or x_k-1	$y_{k+1}=y_k-1$	P_{k+1}
-176	8+1=9	5-1=4	276
276	9	4-1=3	-224
-224	9+1=10	3-1=2	756
756	10	2-1=1	656
656	10	1-1=0	756

Then, (x_{k+1}, y_{k+1}) are the points to be plotted.



#1 Given ellipse parameter with $r_x=8$ and $r_y=6$ use mid-point ellipse drawing to draw an ellipse.

→ Here,

Center = (0,0)

$r_x = 8$ and $r_y = 6$

For region 1,

We start from $(0, r_y) = (0, 6)$

Initially (0,6) was plotted

$P_0 = f(0+1, 6-\frac{1}{2}) = (0+1)^2 \cdot 6^2 + (6-\frac{1}{2})^2 \cdot 8^2 - 8^2 \cdot 6^2$

$= -332$

Now,

P_k	$x_{k+1}=x_k+1$	$y_{k+1}=y_k$ or y_k-1	P_{k+1}	$2r_y^2x_{k+1}$	$2r_x^2y_{k+1}$
-332	0+1=1	6	-224	72	768
-224	1+1=2	6	-44	144	768
-44	2+1=3	6	208	216	768

208	3+1=4	5	-108	288	640
-108	4+1=5	5	288	360	640
288	5+1=6	5-1=4	244	432	512
244	6+1=7	4-1=3	400	504	384

We move out of Region I since $2r_y^2 x_{k+1} > 2r_x^2 y_{k+1}$

For region II,

First point to take/plot is (7,3) since it is the last point of region I.

Initially,

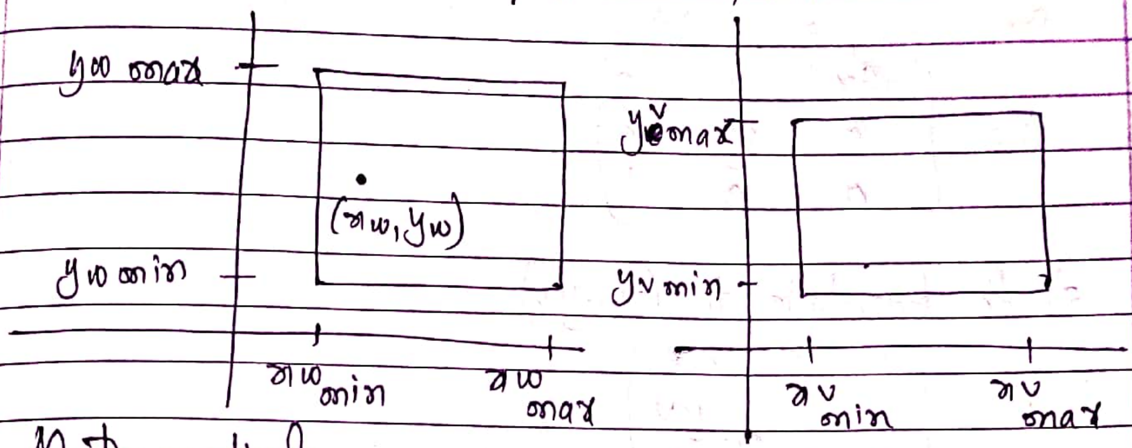
$$P_0 = f(7+\tfrac{1}{2}, 3-1)$$

$$= (7+\tfrac{1}{2})^2 \times 6^2 + (3-1)^2 \times 8^2 - 8^2 \times 6^2 = -23$$

P_k	$x_{k+1} = x_k$ or $x_k - 1$	$y_{k+1} = y_k - 1$	P_{k+1}
-23	7+1=8	3-1=2	$-23+2\times 8\times 6^2 + 8^2\times(1-2\times 2) = 361$
361	8	2-1=1	$361+8^2(1-2\times 1) = 297$
297	8	0	

These (x_{k+1}, y_{k+1}) are the points to be plotted.

Window to Viewport Transformation



Mathematical,

$$\frac{xv - xv_{min}}{xv_{max} - xv_{min}} = \frac{xw - xw_{min}}{xw_{max} - xw_{min}}$$

$$xv - xv_{min} = \frac{xv_{max} - xv_{min}}{xw_{max} - xw_{min}} (xw - xw_{min})$$

$$xv = xv_{min} + \frac{xv_{max} - xv_{min}}{xw_{max} - xw_{min}} (xw - xw_{min})$$

$$xv = xv_{min} + (xw - xw_{min}) \cdot s_x$$

Again,

$$\frac{yv - yv_{min}}{yv_{max} - yv_{min}} = \frac{yw - yw_{min}}{yw_{max} - yw_{min}}$$

$$yv - yv_{min} = \frac{yv_{max} - yv_{min}}{yw_{max} - yw_{min}} (yw - yw_{min})$$

$$yv = yv_{min} + \frac{yv_{max} - yv_{min}}{yw_{max} - yw_{min}} (yw - yw_{min})$$

$$\therefore yv = yv_{min} + (yw - yw_{min}) \cdot s_y$$

Graphical,

1) Translate (xw, yw) to origin

$$T = \begin{bmatrix} 1 & 0 & -xw \\ 0 & 1 & -yw \\ 0 & 0 & 1 \end{bmatrix}$$

2) Perform scaling with S_x and S_y such

$S_x =$, $S_y =$

$$S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3) Re-translate to (x_v, y_v)

$$T^{-1} = \begin{bmatrix} 1 & 0 & x_v \\ 0 & 1 & y_v \\ 0 & 0 & 1 \end{bmatrix}$$

$$CM = T^{-1} \cdot S \cdot T$$

Date
Page

Clipping

Use the clipping method to clip a line starting from A(-13,5) and ending at B(17,11) against the window having its lower corner at (-8,-4) and upper right corner at (12,8)

Soln;

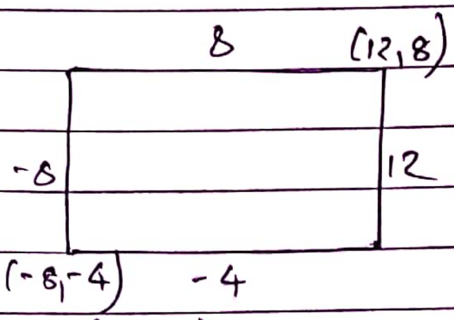
Start point $(x_1, y_1) = A(-13,5)$

End point $(x_2, y_2) = B(17,11)$

For window,

$xw_{min} = -8$, $xw_{max} = 12$,

$yw_{min} = -4$, $yw_{max} = 8$



For A (-13,5)

Bit 1 = 1 (L)

Bit 2 = 0 (R)

Bit 3 = 0 (B)

Bit 4 = 0 (T)

For B (17,11)

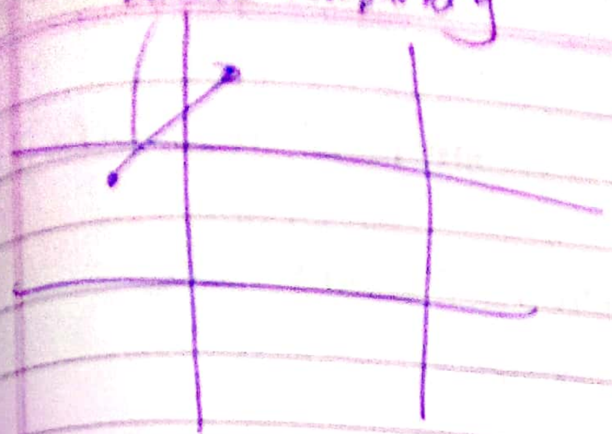
Bit 1 = 0 (L)

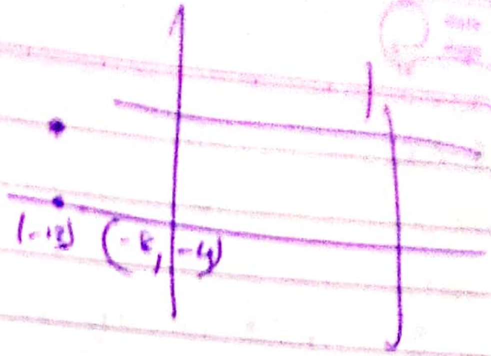
Bit 2 = 1 (R)

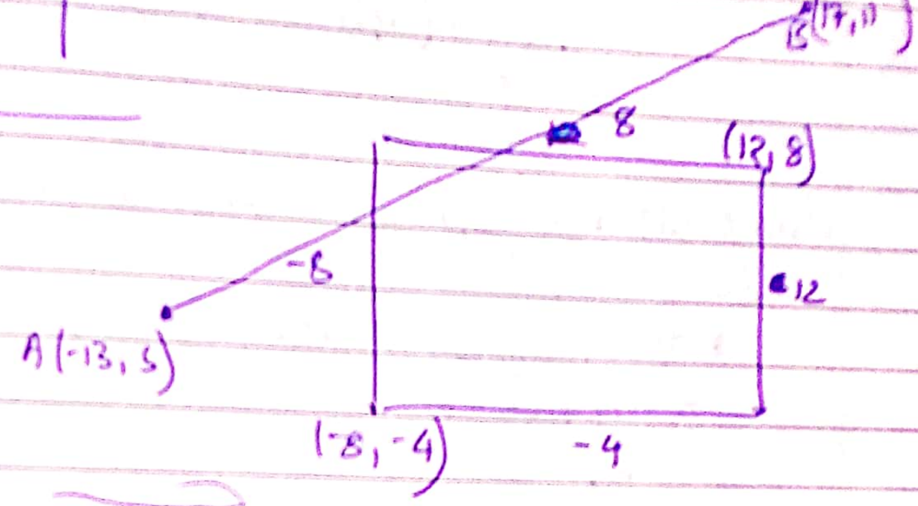
Bit 3 = 0 (B)

Bit 4 = 1 (T)

Outside completely







Solution:

Start point $(x_1, y_1) = A(-13, 5)$

End point $(x_2, y_2) = B(17, 11)$

For window,

$x_{w\,min} = -8$, $x_{max} = 12$, $y_{w\,min} = -4$, $y_{w\,max} = 8$

$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{11 - 5}{17 - (-13)} = \frac{1}{5}$

For A(-13, 5),		For B(17, 11)	
Bit 1 = 1	(L)	Bit 1 = 0	(L)
Bit 2 = 0	(R)	Bit 2 = 1	(R)
Bit 3 = 0	(B)	Bit 3 = 0	(B)
Bit 4 = 0	(T)	Bit 4 = 1	(T)

Logical AND of 0001 and 1010 = 0000 (clipping ~~should be done~~) The line is partially inside and partially outside

For A

Since bit 1 is 1, it intersects with left boundary.

$y = y_1 + m(x - x_1)$ where $x = x_{w\,min} = -8$

$= 5 + m(-8 - (-13)) = \cancel{10} = 5 + \frac{1}{5}[-8 - (-13)] = 6$

A intersects with (-8, ~~10~~)

For B.

Since bit 2 is 1, it intersect with right boundary.

So,

$y = y_1 + m(x - x_1)$ where $x = xw_{max} = 12$

$= 5 + \frac{1}{5}(12 - (-13))$

$= 10$

$\cancel{(12, 30)} = (12, 10)$

Since bit 4 is 1, line intersects with top boundary.

So,

$x = x_1 + \frac{1}{m}(y - y_1)$ where $y = yw_{max} = 8$

$= -13 + \frac{1}{\frac{1}{5}}(8 - 5)$

$= -10$

$\cancel{(-10, 8)} \Rightarrow (2, 8)$

B intersect at $(2, 8)$