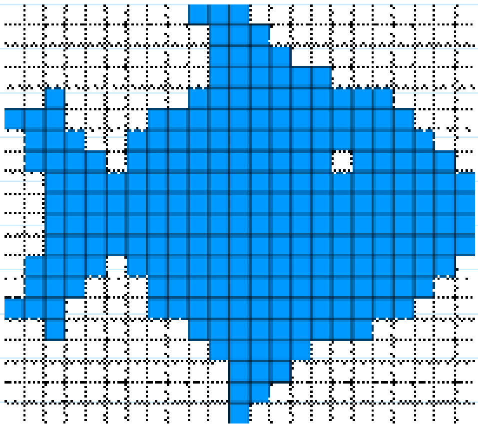
12. Scan conversion algorithms for Line

14 February 2024 12:34

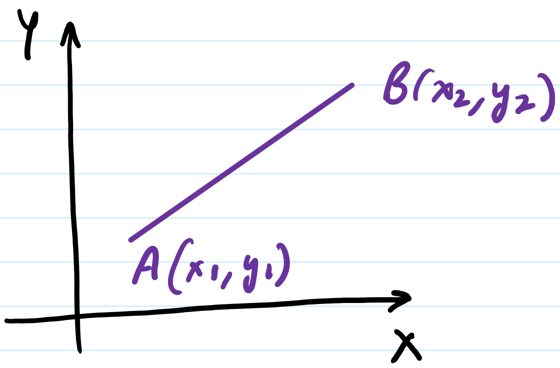
Rasterization / Scan Conversion :-

Representation of continuous graphics objects as a collection of discrete pixels.



Scan Conversion of a Line :-

Line :-



Equation of line,

$$y = mx + c$$

$$m = \Delta y / \Delta x$$

$$\Delta x = x_2 - x_1$$

$$\Delta y = y_2 - y_1$$

Using scan conversion algorithms, line is drawn by plotting the pixels in sequence.

Scan Conversion algorithms for Line Drawing

- Direct Method
- Digital Differential Analyzer (DDA)
- Bresenham's Algorithm

Direct Method :-

Two known end points

↓

Find other points lying on the line using $y = mx + c$

Algorithm :-

① Read $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$

② Calculate

$$dx = x_2 - x_1$$

$dy = y_2 - y_1$

③ Calculate slope

$m = dy/dx$

④ Set (x, y) to starting point

if $dx > 0$ then

$x = x_1$

$y = y_1$

$x_{end} = x_2$

if $dx < 0$ then

$x = x_2$

$y = y_2$

$x_{end} = x_2$

⑤ Now calculate

$c = y - mx$

⑥ Plot a point at current (x, y) coordinates.

⑦ Increment x, $x = x + 1$

⑧ Compute y, $y = mx + c$

⑨ If $x = x_{end}$ then stop

otherwise Go to ⑥

Q8- Draw a line using direct method between points (0,0) & (8, 16)

① (x_1, y_1) (x_2, y_2)

(0,0) and (8, 16)

② $dx = 8-0 = 8$, $dy = 16-0 = 16$.

③ $m = dy/dx = 16/8 = 2$.

④ Set initial point (x, y)

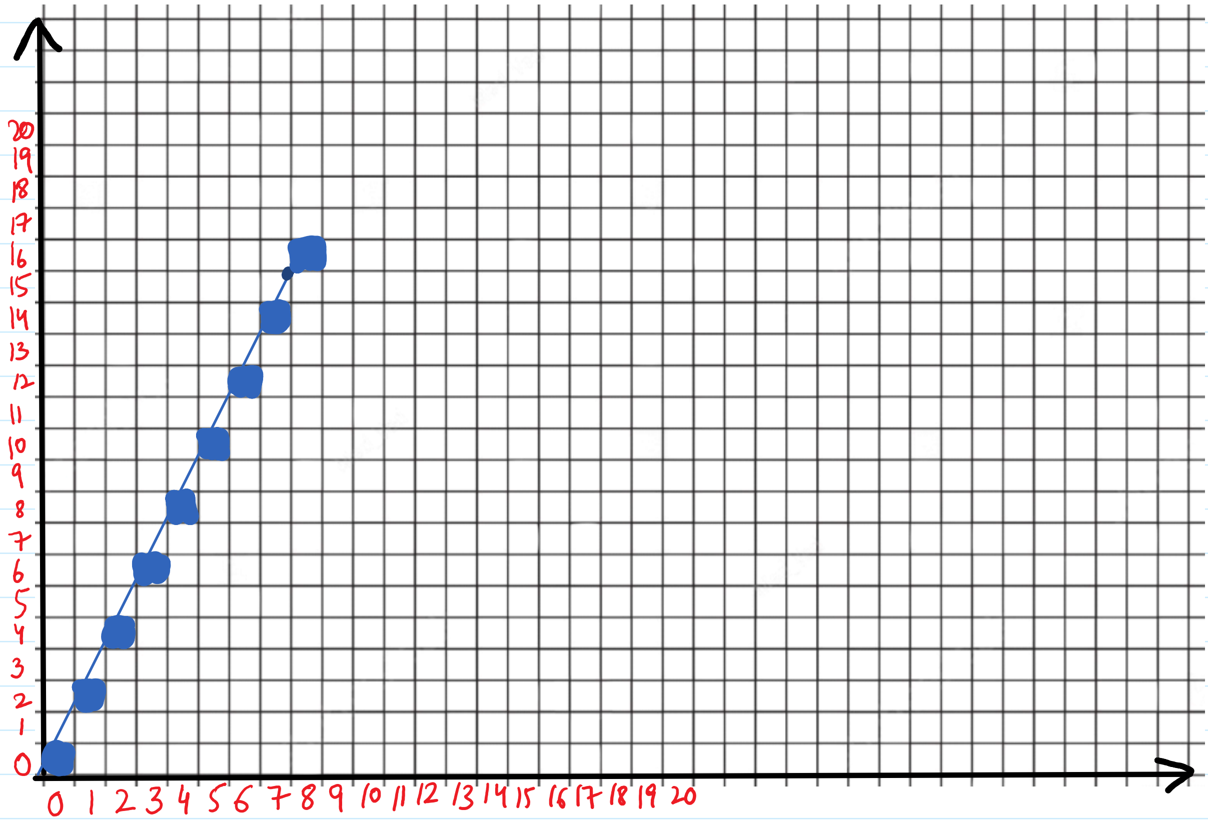
$dx > 0$

$\Rightarrow x = 0$ $y = 0$ $X_{end} = 8$

⑤ Calculate $c = y - mx$ $c = 0$

While $x = x_{end}$

$x = x+1$	$y = mx + 0$	Points
0	0	$P_1(0,0)$
1	2	$P_2(1,2)$
2	4	$P_3(2,4)$
3	6	$P_4(3,6)$
4	8	$P_5(4,8)$
5	10	$P_6(5,10)$
6	12	$P_7(6,12)$
7	14	$P_8(7,14)$
8	16	$P_9(8,16)$



② Digital Differential Analyzer (DDA)

① $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$

② Finding appropriate length of the line

if (abs $(x_2 - x_1)$ > abs $(y_2 - y_1)$) then

length = abs $(x_2 - x_1)$

otherwise

length = abs $(y_2 - y_1)$

③ Find raster unit

$dx = (x_2 - x_1)/\ length$

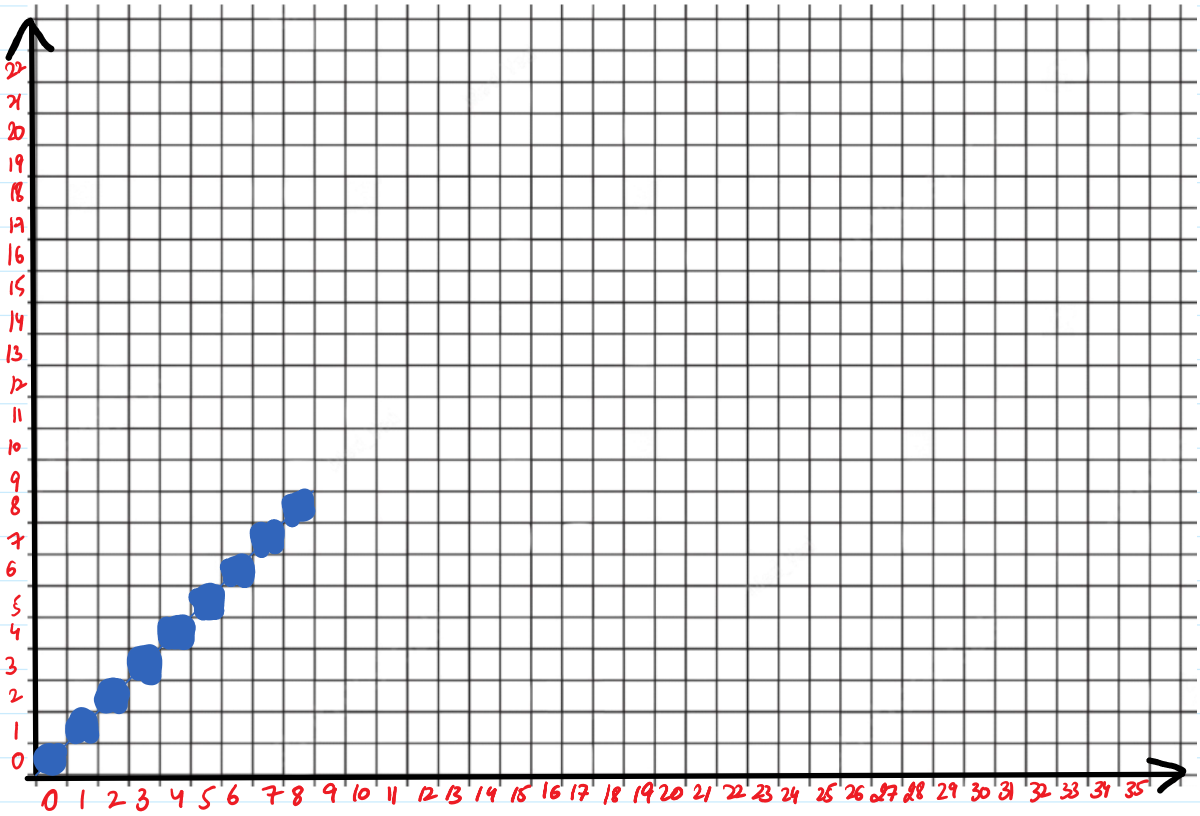
$dy = (y_2 - y_1)/\ length$

4. set $x = x_1$, $y = y_1$ and $i=0$
5. Plot (x, y)
   $x = x + dx$
   $y = y + dy$
6. Repeat 5 until $i <= length$

Example :- $(0,0)$ & $(8,8)$

1. $x_1 = 0$ $y_1 = 0$ $x_2 = 8$ $y = 8$
2. $abs\ (x_2 - x_1) = 8$
   $abs\ (y_2 - y_1) = 8$
   $length = 8$
3. $dx = 1$
   $dy = 1$
4. Set (x, y) $x=0$, $y=0$
5. Execute until $i <= length$

i	x	y	Points
0	0	0	(0,0)
1	1	1	(1,1)
2	2	2	(2,2)
3	3	3	(3,3)
4	4	4	(4,4)
5	5	5	(5,5)
6	6	6	(6,6)
7	7	7	(7,7)
8	8	8	(8,8)



"Floating Point Calculations"

③ Bresenham's Line Algorithm

Jack Elton Bresenham, 1962
(IBM)

Using only integer calculations.

Algorithm :—

① $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$

② Calculate

$$dx = x_2 - x_1$$
$$dy = y_2 - y_1$$

③ Calculate decision parameter P

$$P = 2dy - dx$$

④ Set initial point

$$x = x_1, \quad y = y_1$$
and $i = 0$

⑤ Plot (x, y)

if $P < 0$ then

$x = x+1$

$p = p + 2dx$

else

$x = x+1$

$y = y+1$

$p = p + 2dy - 2dx$

(6) Repeat (5) until $i <= dx$

Example :- Draw $(20,10)$ and $(30,18)$

(1) $(20,10)$ & $(30,18)$

$x_1 = 20 \quad y_1 = 10 \quad x_2 = 30 \quad y_2 = 18$

(2) $dx = x_2 - x_1 = 10$

$dy = y_2 - y_1 = 8$

(3) Decision Parameter

$p = 2dy - dx$

$= 6$

(4) Set (x, y)

$x = x_1 = 20$

$y = y_1 = 10$

and $i = 0$

(5) Execute until $i \leq dx$

i	p	Points
0		(20, 10)
1	6	(21, 11)
2	2	(22, 12)
3	-2	(23, 12)
4	14	(24, 13)
5	10	(25, 14)
6	6	(26, 15)
7	2	(27, 16)
8	-2	(28, 16)
9	14	(29, 17)
10	10	(30, 18)