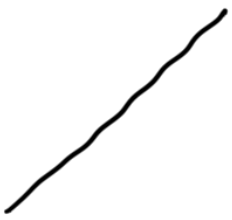
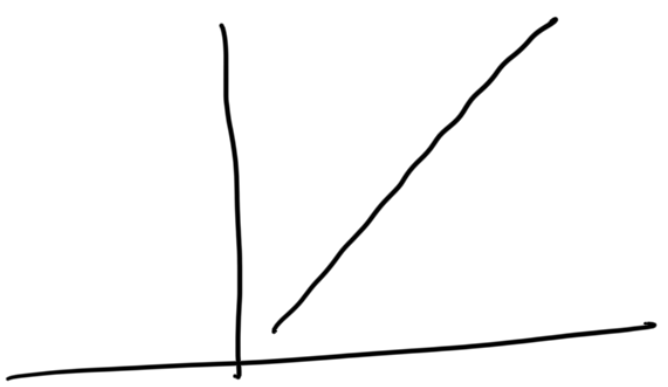
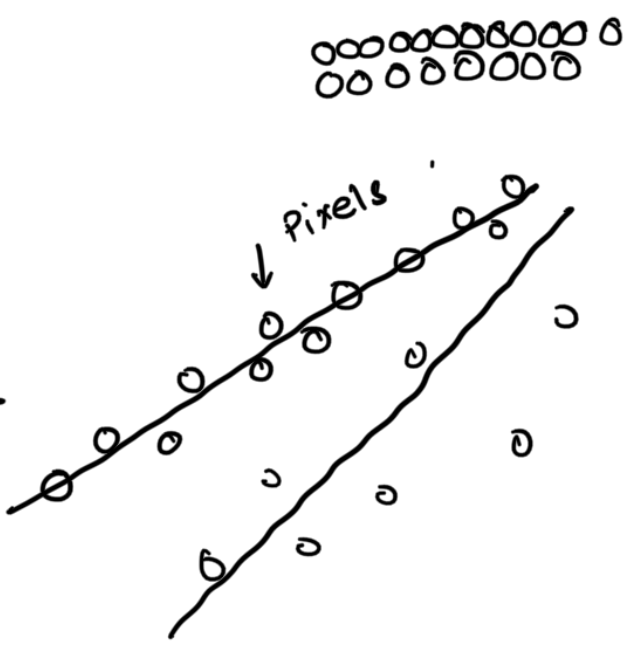
CSE423 Week 1



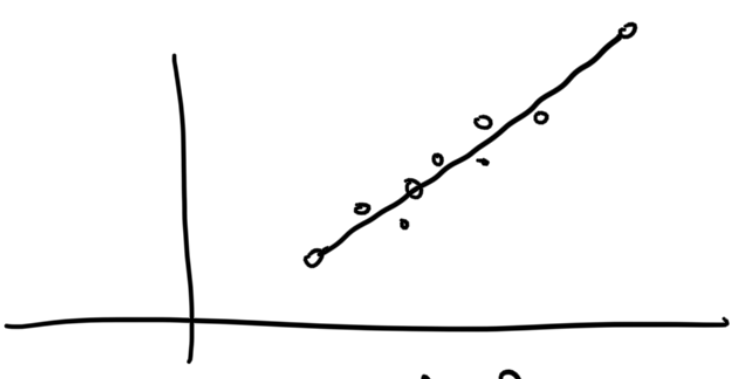
Line Drawing Algorithm

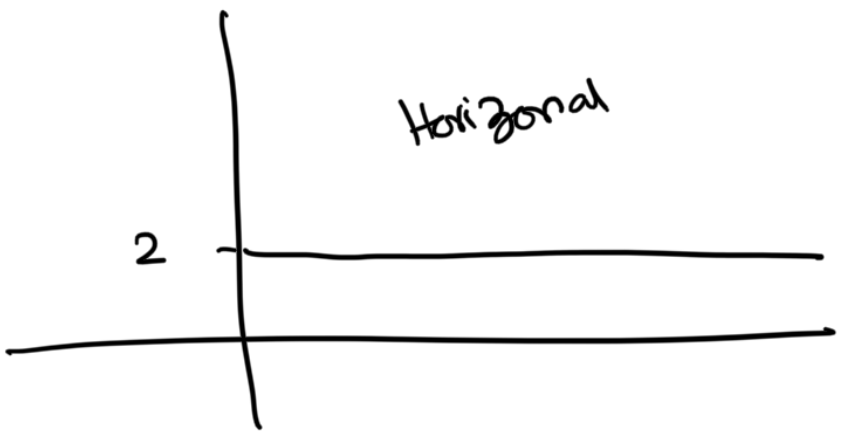
- DDA
- Midpoint
- Midpoint with eight way symmetry

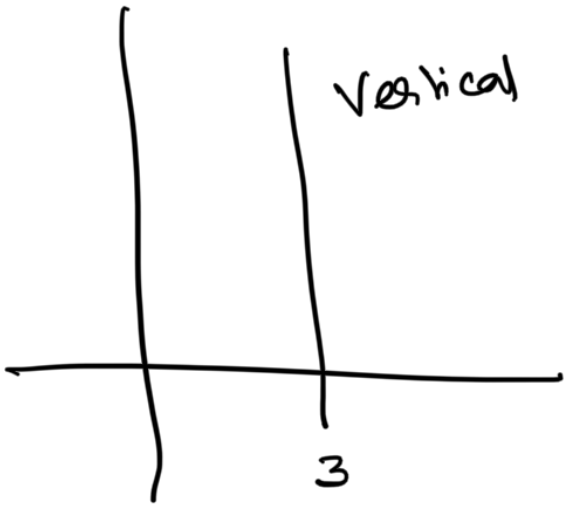




Simple Approach







Target → To find out all the necessary pixels to draw a line

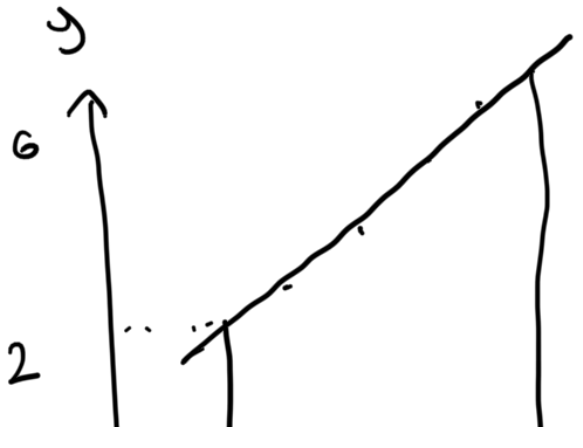
(x_1, y_1) & (x_2, y_2)

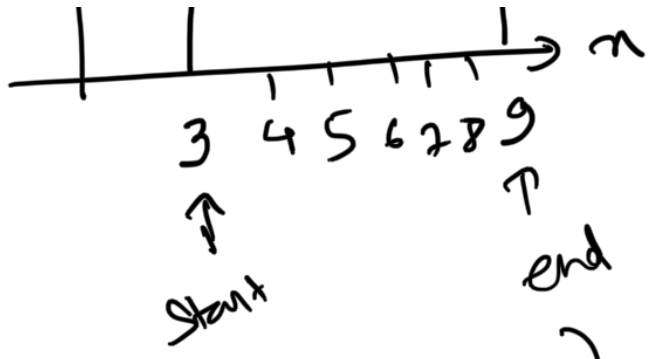
$(3, 2)$ & $(9, 6)$

$y = mx + c$

↓ gradient

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





$$m = \frac{6-2}{9-3} = \frac{4}{6} = \frac{2}{3}$$

(3, 2) (1000, (n))

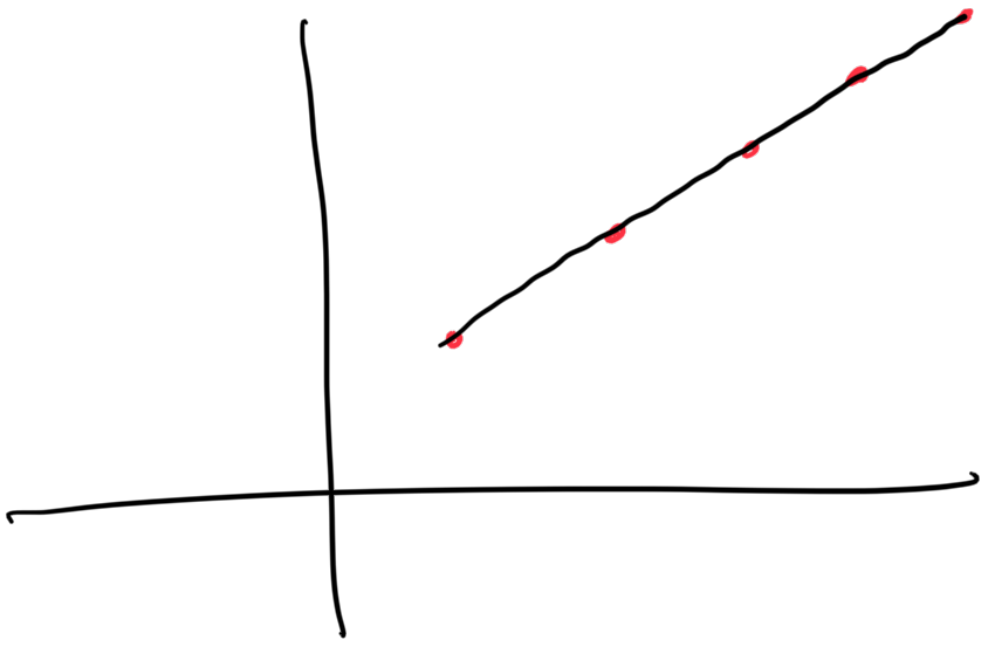
$$y - y_1 = m(x - x_1)$$

$$y - 2 = \frac{2}{3}(x - 3)$$

$$y = \frac{2}{3}x - 2 + 2$$

$$y = \frac{2}{3}x$$

x	y		rounding	
3	$y = \frac{2}{3} \times 3$	$= 2$	≈ 2	(3, 2)
4	$y = \frac{2}{3} \times 4$	$= 2.7$	≈ 3	(4, 3)
5	$y = \frac{2}{3} \times 5$	$= 3.3$	≈ 3	(5, 3)
6	$y = \frac{2}{3} \times 6$	$= 4$	≈ 4	(6, 4)
7	$y = \frac{2}{3} \times 7$	$= 4.6$	≈ 5	(7, 5)
8	$y = \frac{2}{3} \times 8$	$= 5.3$	≈ 5	(8, 5)
9	$y = \frac{2}{3} \times 9$	$= 6$	≈ 6	(9, 6)



Problem !

→ slow

→ Continuous y ber kurte hai

DDA Algorithm

↳ Digital Differential Analyzer.

If m is between $-1 < m < 1$

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

else,

$$y_{k+1} = y_k + 1$$
$$x_{k+1} = x_k + 1/m$$

Then, round off to the nearest pixel.

For example 1

(2,2) ——— (7,5)

$$m = \frac{5-2}{7-2} = \frac{3}{5} = 0.6$$

$-1 < m < 1$

$x = 2+1 =$

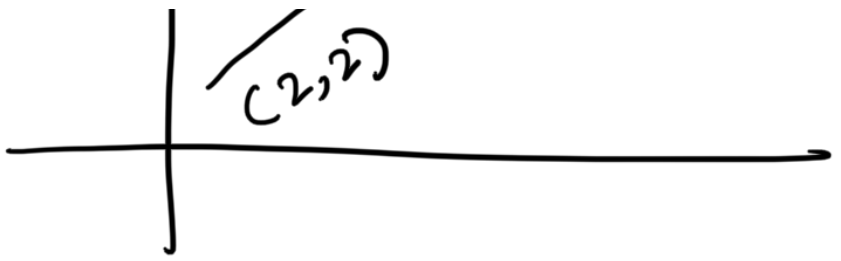
x	y	y (round off)	Pixel
2	2	2	(2,2)
3	2+0.6 = 2.6	3	(3,3)
4	2.6+0.6 = 3.2	3	(4,3)
5	3.2+0.6 = 3.8	4	(5,4)
6	3.8+0.6 = 4.4	4	(6,4)
7	4.4+0.6 = 5	5	(7,5)

End

Example 2

(5,7).

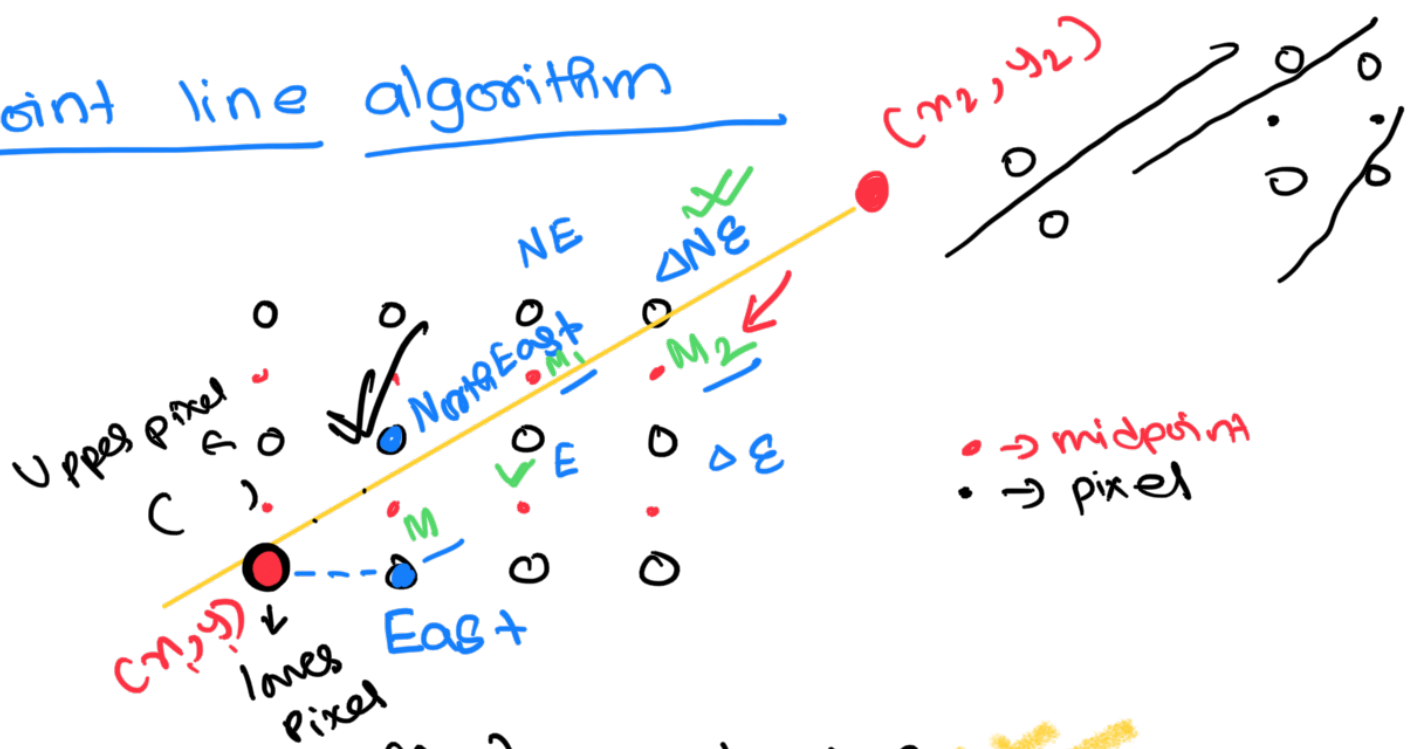
|



$$m = \frac{7-2}{5-2} = \frac{5}{3} = 1.7$$

y	n	n (round off)	pixel
2	2	2	(2,2)
3 (2+1)	$2+\frac{1}{3}=2.58$	3	(3,3)
4	$2.58+.58=3.16$	3	(3,4)
5	$3.16+.58=3.74$	4	(4,5)
6	$3.74+.58=4.32$	4	(4,6)
7 (voilà →)	$4.32+.58=4.9$	5	(5,7)

Midpoint line algorithm



$$f(n,y) = an + by + c$$

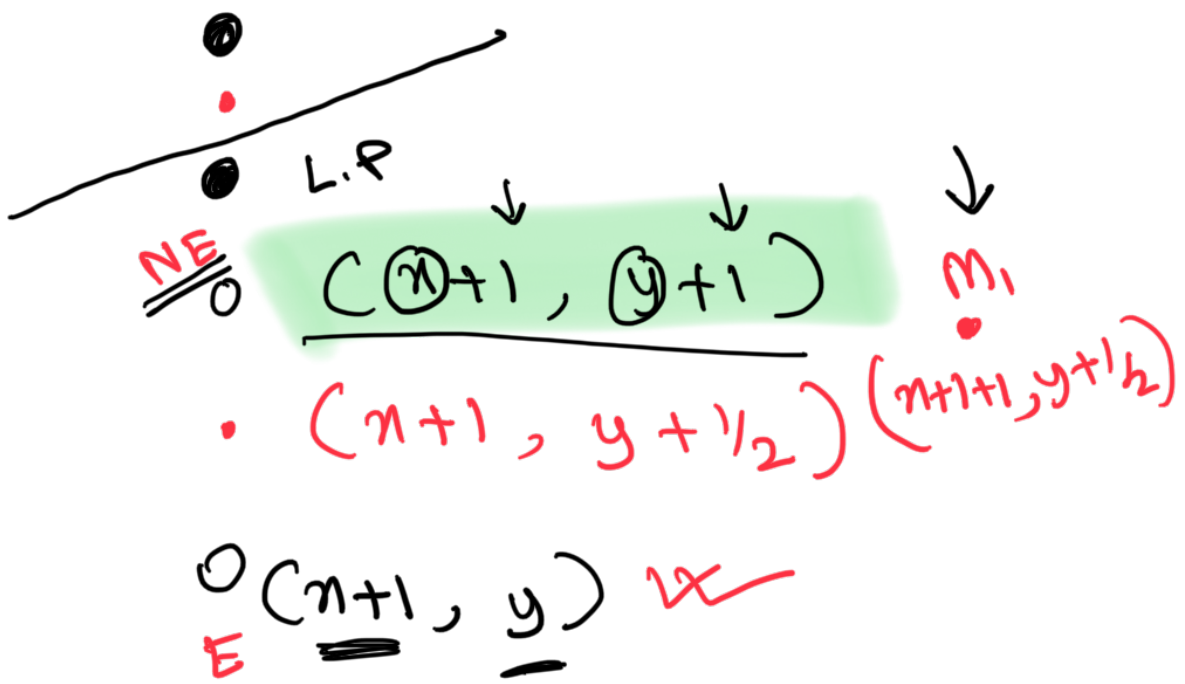
Conditions :

dold = f(Midpoint)	Midpoint's Position	Pixel
① -ve	Upperside of line	lower Pixel (E)

② +ve ✓

lower side of line

Upper Pixel (NE) ✓



$$y = mx + c \quad \cdots (1) \qquad m = \frac{\Delta y}{\Delta x}$$

$$y = \frac{\Delta y}{\Delta x} x + c$$

$$\Delta x\, y - \Delta y\, x - \Delta x\, c = 0 \quad (x-1)$$

We know,

$$\Delta y\, x - \Delta x\, y + \Delta x\, c = 0$$

$$a x + b y + c = 0$$

$$a = \Delta y$$
$$b = -\Delta x$$
$$c = \Delta x$$

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

x: x+1, x+2 ; y: y+1/2, y+1/2

$$f(M)\,/\,d_{old} = f\left(x+1, y+\tfrac{1}{2}\right)$$

$$d_{old} = a(x+1) + b\left(y+\tfrac{1}{2}\right) + c$$

M initial ⇒

Case 1: Upper Pixel (NE)

$$\left(\frac{x+1+1}{x+2}, \frac{y+\frac{1}{2}+1}{y+\frac{1}{2}+1}\right)$$

$f(M_1) = (x+2, y+3/2)$

$d_{new} = a(x+2) + b(y+3/2) + c$

$d_{new} - d_{old} = a(x+2) + b(y+3/2) + c -$
$a(x+1) - b(y+1/2) - c$

$= ax + 2a + by + 3/2\,b + c -$
$ax - a - by - 1/2\,b - c$

$d_{new} - d_{old} = a + b$

$d_{new} = d_{old} + a + b$ — ΔNE

Case 2 : Lower Pixel / E

$f(M_1) / d_{new} = f(x+1+1, y+1/2)$

$= f(x+2, y+1/2)$

$d_{new} = a(x+2) + b(y+1/2) + c$

$d_{new} - d_{old} = a(x+2) + b(y+1/2) + c$
$- a(x+1) - b(y+1/2) - c$

$d_{new} - d_{old} = ax + 2a + by + 1/2\,b + c$
$- ax - a - by - 1/2\,b - c$

$d_{new} = d_{old} + a$ — ΔE

Initially, $m = (x+1, y+1/2)$ (x,y)

$d_{old} = a(x+1) + b(y+1/2) + c$

$= ax + a + by + 1/2\,b + c$

$= ax + by + c + a + 1/2\,b$

$$d_{old} = 0 + a + \tfrac{1}{2}b$$

$$d_{old} = a + \tfrac{1}{2}b$$ (Multiply by 2)

we need to convert to integers from floating point

$$d_{ow} = 2a + b$$

$$d_{old} = 2\Delta y - \Delta x$$

$a = \Delta y$

$b = -\Delta x$

V.P / ΔNE

$$d_{new} = d_{old} + \underbrace{2\Delta y - 2\Delta x}_{\Delta NE} \quad \text{if}$$

L.P / ΔE

$$d_{new} = d_{old} + \underbrace{2\Delta y}_{\Delta E} \quad \text{if}$$

$$d_{NE} = 2(\Delta y - \Delta x)$$

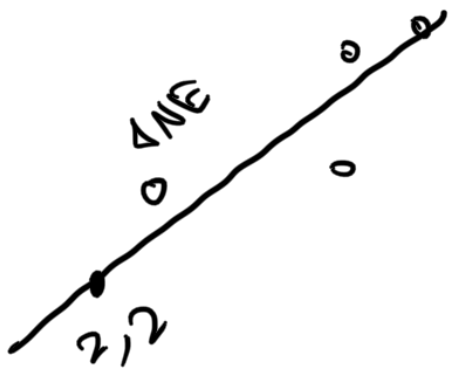
$$d_{\Delta E} = 2\Delta y$$

$$d_{old} = 2\Delta y - \Delta x$$

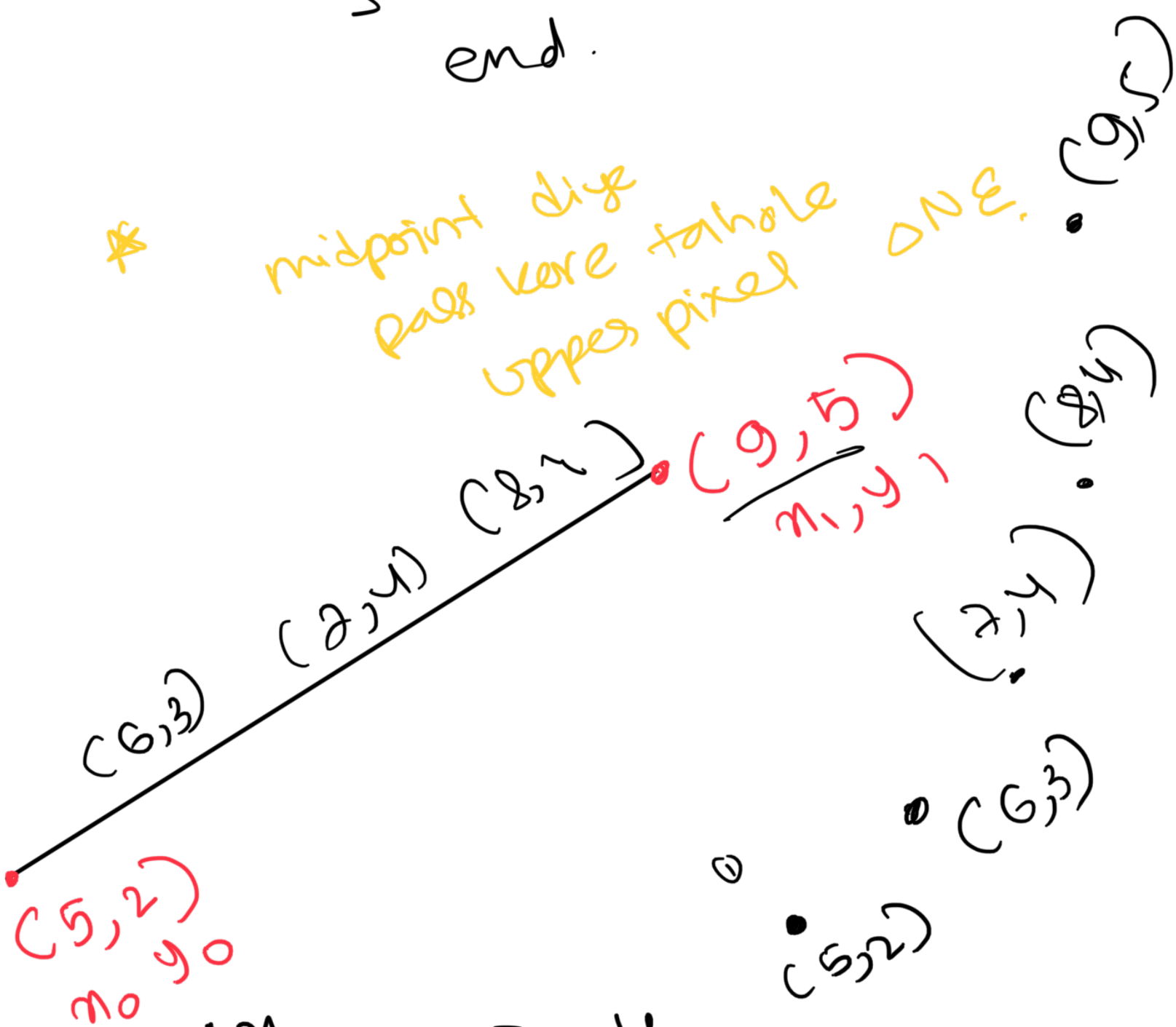
```
Void draw_line(int x0, int y0, int x1, int y1)
    int Δx = x1 - x0 ;
    int Δy = y1 - y0 ;
    int d_old = 2Δy - Δx ;
    int dE = 2 * Δy ;
    int dNE = 2 * (Δy - Δx) ;
    int x = x0 , int y = y0
    draw pixel (x, y) ;
    while (x < x1) {
        if (d_old < 0) {        ΔE
            x++ ;
```

(2,2)

```
        dold = dold + dE ;
    else {                    ΔNE
        x++ ;
        y++ ;
        dold = dold + dNE ;
    }
    drawpixel (x , y) ;
}
}
end.
```



* midpoint dig gak kere tahole ΔNE. (9,5)



$\Delta x = 9 - 5 = 4$

$\Delta y = 5 - 2 = 3$

$d_{old} = 2 \times 3 - 4 = 2$ ✗

$dE = 2 \times 3 = 6$ ←

$d_{NE} = 2 \times (3-4)$

$= -2 \times e$

$x = 5 , \quad y = 2$

$(5,2)$ Pixel

$5 < 9$

① ΔNE

$x = 5+1 = 6$

$y = 2+1 = 3$

$d_{old} = 2-2 = 0$

$(6,3)$ Pixel.

$6 < 9$

② ΔNE $d_{old} = 0$ or $d > 0$ NE

$x = 6+1 = 7$

$y = 3+1 = 4$

$d = 0-2 = -2$ W

$(7,4)$ Pixel.

③ ΔE $7 < 9$

$x = 7+1 = 8$

$d = -2+6 = 4$

(8, 4) (Pixel)

$8 < 9$

(4) ΔNE

$n = 8 + 1 = 9$

$y = 5$

$d = 4 - 2 = 2$

(9, 5) pixel.