Computer Graphics

Date- 14/3/2

Practical : (i) line / Circle generation
(ii) Rotation (Δ) matrix given.

Memory = frame buffer

Computer graphics :- ① Creating objects, ② Storing the objects (3) Manipulation of still objects

Memory → frame buffer

(user interation)

Types:- ① Interactive graphics → Playing game.
② Passive graphics. → T.V. cat cartoon.
(no user interaction)

2-marks

☆ Applications :- Game playing, Presentation, marketing advertisement, medical diagonis, business. education, all types of reports, Air craft (learning purpose), Architecture

☆ Line generation algorithm :-

(i) DDA (Digital differential Analyzer)
(Difference b/w points)

→ It is an incremental algorithm (means progresive)
If two point is given than how to calculate.

Δx & Δy

if $P_1(x_1, y_1) \longrightarrow (1,3)$

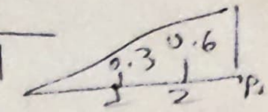
$P_2(x_2, y_2) \longrightarrow (5,8)$

$\therefore \Delta x = x_2 - x_1 \Rightarrow 5-1 \Rightarrow 4$

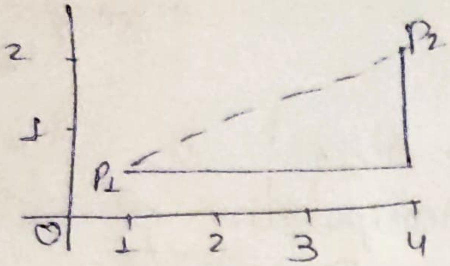
$\Delta y = y_2 - y_1 \Rightarrow 8-3 \Rightarrow 5.$

Slope ⇒ कितना ratio में बढ़ता है

Slope $m = \frac{y_2 - y_1}{x_2 - x_1} \Rightarrow \frac{\Delta y}{\Delta x} \Rightarrow \frac{5}{4}$



Eg. (I)



Than $P_1(1,1)$ & $P_2(4,2)$

$\therefore \Delta x = 4 - 1 \Rightarrow 3$

$\Delta y = 2 - 1 \Rightarrow 1.$

$\therefore m = \frac{\Delta y}{\Delta x} = \frac{1}{3} \Rightarrow 0.33$

(Δx & Δy)

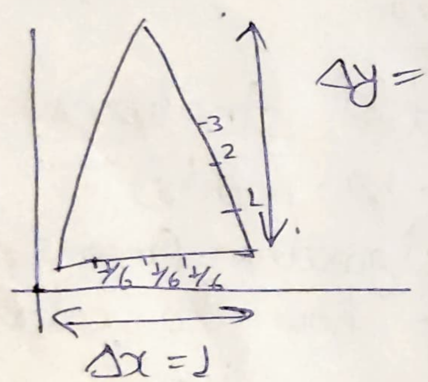
means जो सबसे ज्यादा वो as it is बढ़ेगा और उसका दूसरा उसके proportional के हिसाब से बढ़ेगा।
means $\Delta x^{(3)}$ value ज्यादा है वो same और $\Delta y^{(1)}$ उसके $\frac{1}{3}$ value से बढ़ेगा।

means अगर $\Delta x = 3 \longrightarrow 1$ unit

$\Delta y = 1 \longrightarrow \frac{1}{3}$ unit.

Similarly reverse is also possible.

Eg. (II)



$\Delta y = 6.$

means ~~Δx~~ $\frac{1}{6}$ unit बढ़े

2-marks

पहले slope calculate फिर नीचे का conditions

★ DDA

Step 1 Calculate slope m.

(a) if $|m| < 1$ eg. $m = \frac{1}{3} = 0.3332$

than $\boxed{\begin{aligned} x &= x + 1 \\ y &= y + m \end{aligned}}$

(b) if $|m| > 1$ eg: $m = \frac{6}{1} \Rightarrow 6 > 1$

than $\boxed{\begin{array}{l} y = y + 1 \\ x = x + \frac{1}{m} \end{array}}$

(6,6)

eg: (1,1)

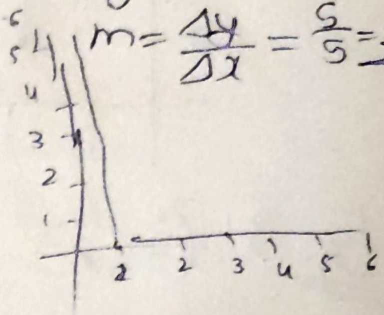
$\Delta x = 6 - 1 = 5$

$\Delta y = 6 - 1 = 5$

$m = \frac{\Delta y}{\Delta x} = \frac{5}{5} = 1$

(c) if $m = 1$.

than $\boxed{\begin{array}{l} x = x + 1 \\ y = y + 1 \end{array}}$



Eg. ⇒ Draw a line by using DDA.
(2,1) to (8,5) using DDA.

$\Delta x = 8 - 2 \Rightarrow 6$.

$\Delta y = 5 - 1 \Rightarrow 4$

$m = \frac{\Delta y}{\Delta x} \Rightarrow \frac{4}{6} \Rightarrow \frac{2}{3} = 0.6667$

Case ① ~~0.66~~ < 1.

$\therefore$ $x \neq$ $y =$

Round Next based on the 3 condition

x	y	$R(y)$
Initially starting 2	1.	→ 1
3	(1.6667)	→ 2
4	(1.6667 + 0.6667) = 2.334	→ 2
5	3	→ 3
6	3.6667	→ 4
7	4.3334	→ 4.
8	5	→ 5.

here. ⊙ $0.6667 < 1$.

$\therefore \boxed{\begin{array}{l} x = x + 1. \\ y = y + m. \end{array}}$

↓

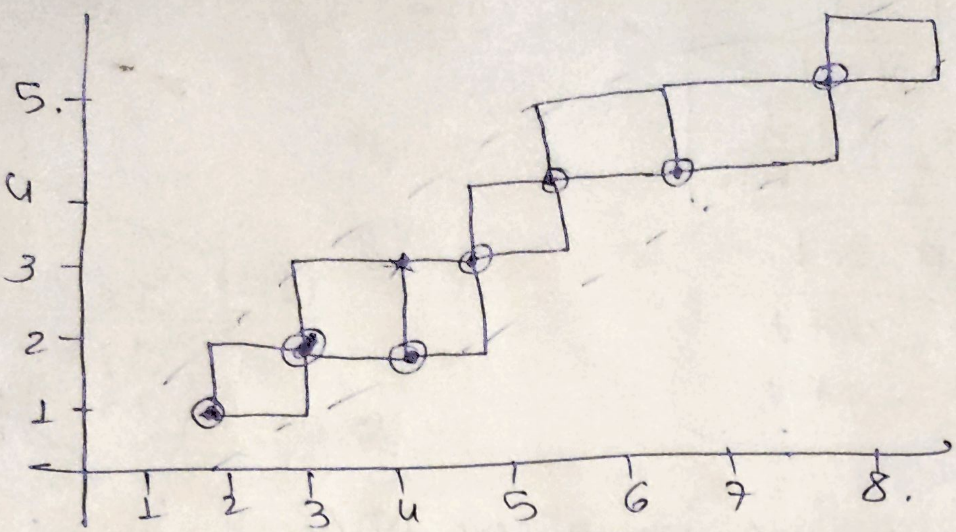
x will increase until last point (8).

(ii) $y = 1 + 0.6667$.

(iii)

• .5 → uppervalue

• .5 से नीचे → lowervalue



Eg. ⇒ $(\overset{x_1}{3}, \overset{y_1}{2})$ to $(\overset{x_2}{7}, \overset{y_2}{8})$

$\Delta x = 7 - 3 \Rightarrow 4$

$\Delta y = 8 - 2 \Rightarrow 6.$

$m = \frac{\Delta y}{\Delta x} \Rightarrow \frac{6}{4} \Rightarrow 1.5$

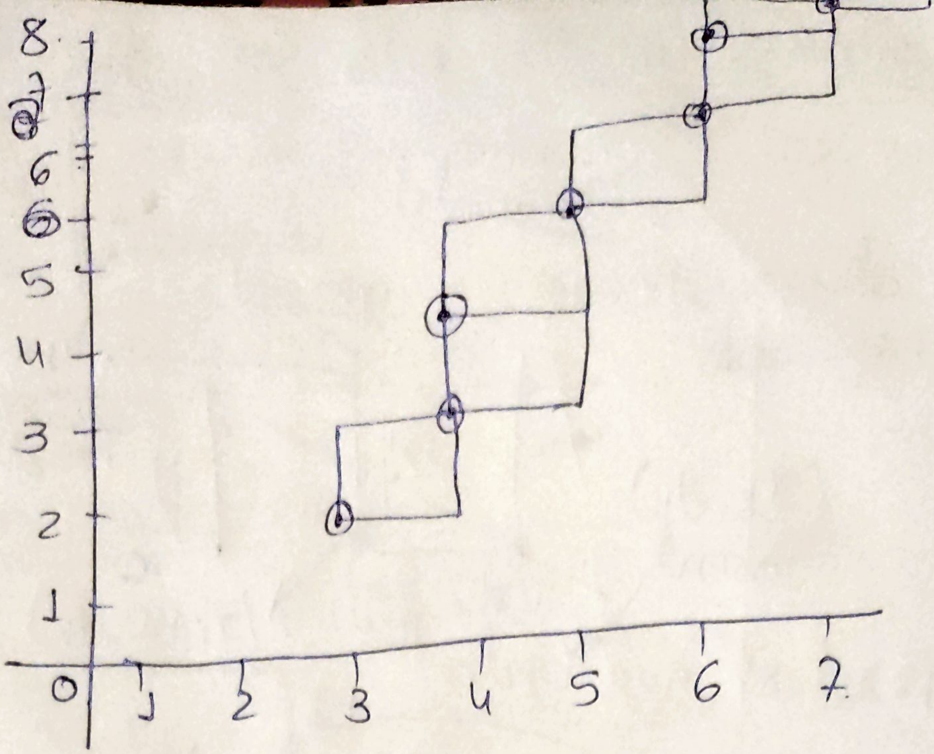
$\boxed{\frac{1}{m} = 0.6667}$

here. $1.5 > 1$

$\therefore\ y = y + 1$ & $x = x + \frac{1}{m}.$

जो ज्यादा है वो +1
बढ़ते जायेगा
जैसे Δy ज्यादा यहाँ

x		R(x)	y.
3	→	3.	2
3.6667	→	4	3
4.3334	→	4	4
5	→	5	5
5.6667	→	6	6
6.3334	→	6	7
7	→	7.	8.



Eg. (12, 12) to (19, 19).

$\Delta x = 19 - 12 \Rightarrow 7$ $\boxed{m = 1}$

$\Delta y = 19 - 12 \Rightarrow 7$

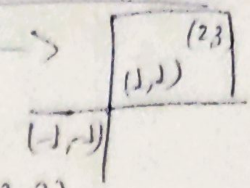
$\therefore$ $m = 1$

than $x = x + 1$
$y = y + 1$

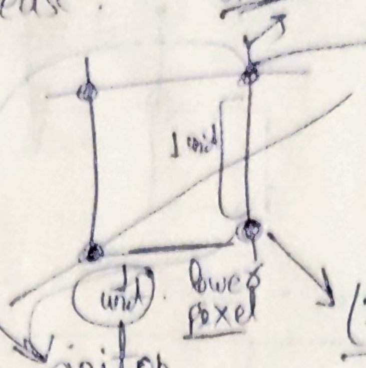
x	y
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19

Bresenhams line generation Algorithm (Date- 15/3/2…)

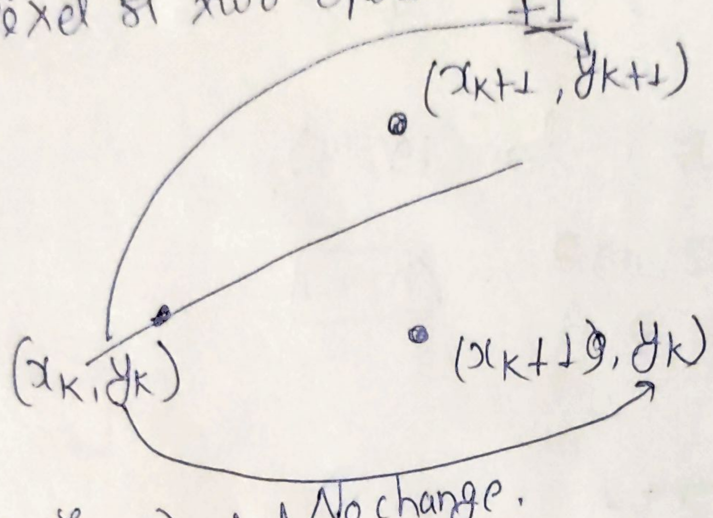
It makes in 1st corner (quardrants)



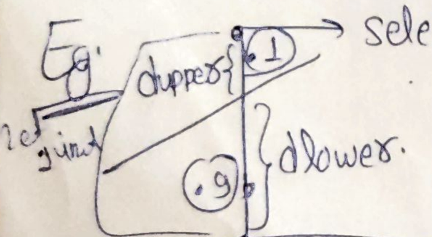
x is always increase.

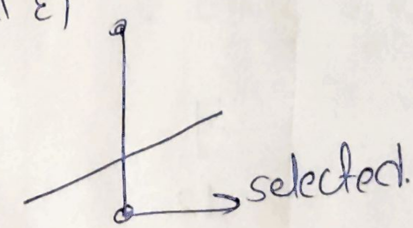


Next pixel के two opiton



जो line पास है, उसे select करना है।

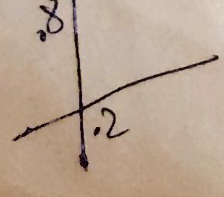




$\boxed{\text{dlower} - \text{dupper}} = 0.9 - 0.1 = .08$ (+ve) OR 0

> 0 या +ve → upper pixel select → upper pixel select
-ve → lower pixel select.

Eg. (II)



dlower - dupper ⇒ 0.2 - 0.8 ⇒ -0.6

lower pixel selected

Drawing a line by using BLG Algo.

Step 1. Initial Decision Par: $P_k = 2\Delta y - \Delta x$ → provide ② value

(a) ≤ 0, than $x_k = x_{k+1}$, y_k

$y_k = y_k$.

(b) ≥ 0

$x_k = x_{k+1}$,

$y_k = y_{k+1}$

② decision parameter (P)

Negative

For (a) ≤ 0: $$d_{k+1} = P_k + 2\Delta Y$$

+ve

For (b) ≥ 0: $$d_{k+1} = P_k + 2\Delta Y - 2\Delta X$$

Eg:- (2, 1) to (8, 5)

$\Delta y = 5 - 1 = 4$ & $\Delta x = 8 - 2 = 6$.

$2\Delta x = 2(6) \Rightarrow 12$

For. (साथ में)

$2\Delta y = 2(4) \Rightarrow 8$

$2\Delta y - 2\Delta x = 8 - 12 \Rightarrow \underline{-4}$

$P_k \Rightarrow$ previous parameter

x	y	$P_k (2\Delta y - \Delta x)$ First time	
2	1	$8 - 6 \Rightarrow 2$	+ve. ← ($d_{k+1} = P_k + 2\Delta y - 2\Delta x$)
3	1+1 ⇒ 2 (step 1 apply)	$2 - 4 \Rightarrow -2$	(-ve)
4	2	$-2 + 8 \Rightarrow 6$	(+ve) (y is incre. +1)
5	3	$6 - 4 \Rightarrow 2$	(+ve)
6	4	$2 - 4 \Rightarrow -2$	(-ve) (y is No change) → ($d_{k+1} = P_k + 2\Delta y$)
7	4	$-2 + 8 \Rightarrow 6$	(+ve)
8	5	~~6 - 4 ⇒ 2 (+ve)~~	last step stop

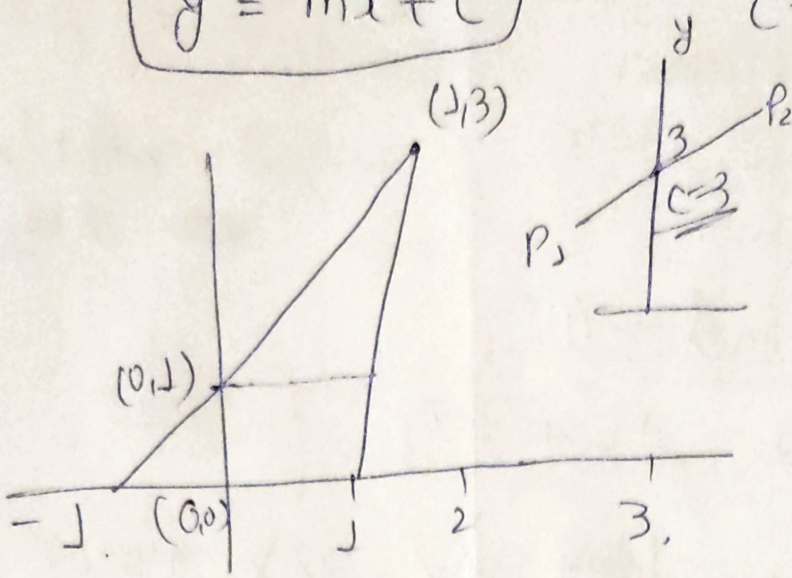
+ve means y_{k+1} & -ve mean y_k.

Equation of line :- $(y_k + 1)$

Date - 16/3/22

$$\boxed{y = mx + C}$$

$C = y$ intercept.
(line y को कहा काट रहा)



$\underline{C = y_1}$

here $C = 1$ & $m = \frac{y_2 - y_1}{x_2 - x_1} \Rightarrow \frac{2}{1} = 2.$

Let $(1, 3)$ means $x = 1$ than find y?

$y = mx + C \Rightarrow y = 2(1) + 1 \Rightarrow \boxed{y = 3}.$

10 marks

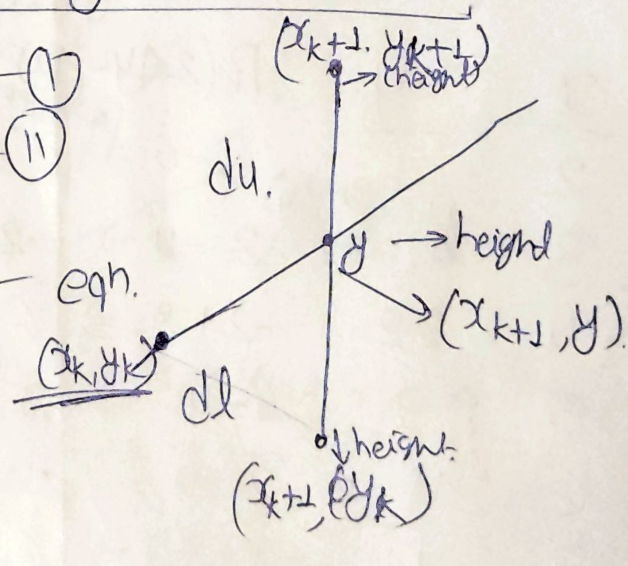
Derivation by using $mx + C$:-

$\boxed{dl\ (dlower) = y - y_k}$ — (I)

$\boxed{du = y_k + 1 - y}$ — (II)

Common formula :-

$\boxed{y = mx + C}$ — eqn.



actual y value

$y = m(x_k + 1) + C$

From eqn (I).

$dl = m(x_k + 1) + C - y_k.$

From eqn (II). $du = y_k + 1 - [m(x_k + 1) + C]$

$\therefore\ dl - du = [m(x_k+1)+c-y_k] - [y_k+1-[m(x_k+1)+c]]$

$dl - du = m(x_k+1)+c-y_k - y_k - 1 + m(x_k+1)+c$

$= 2m(x_k+1) - 2y_k + 2c - 1$

$dl-du = 2\dfrac{\Delta y}{\Delta x}(x_k+1) - 2y_k + 2c - 1.$

multiply Δx in both side.

$\Delta x(dl-du) = 2 * \Delta x * \dfrac{\Delta y}{\Delta x}(x_k+1) - 2\Delta x y_k + 2\Delta x c - \Delta x$

here $\Delta x(dl-du)$ = decision parameter (d_k)

$d_k = 2\Delta y(x_k+1) - 2\Delta x y_k + 2\Delta x c - \Delta x$ — (iii)

$= 2\Delta y x_k + 2\Delta y - 2\Delta x y_k + 2\Delta x c - \Delta x$

here $2\Delta y(x_k) \rightarrow$ change means बढ़ते जाएगा

same $\cdot\ 2\Delta x(y_k)$

here $2\Delta y + 2\Delta x c - \Delta x \rightarrow$ constant (b)

previous

$$\therefore\ \boxed{d_k = 2\Delta y x_k - 2\Delta x y_k + b.} \text{ — (iv)}$$

x_{k+1}

next value

x_k = previous val

next vals

$d_{k+1} = 2\Delta y x_{k+1} - 2\Delta x y_{k+1} + b$ — (v).

$d_{k+1} - d_k = 2\Delta y x_{k+1} - 2\Delta x y_{k+1} + \cancel{b} -$
$2\Delta y x_k + 2\Delta x y_k - \cancel{b}$

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

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

Next we have to check condition $<, \geq$ y_{k+1} y_k

For lower pixel. (-ve, < 0)

$$d_{k+1} - d_k = 2\Delta y - 2\Delta x(y_k - y_k)$$

$$d_{k+1} - d_k = 2\Delta y$$

$\boxed{\text{Put } y_{k+1} = y_k}$

$$\boxed{d_{k+1} = d_k + 2\Delta y}$$

For upper pixel (+ve, ≥ 0)

Put ~~$y_{k+1} = y_k+1$~~ y_k+1.

$$d_{k+1} - d_k = 2\Delta y - 2\Delta x(y_{k+1} - y_k)$$

$$= 2\Delta y - 2\Delta x(y_k + 1 - y_k)$$

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

$$d_{k+1} - d_k = 2\Delta y - 2\Delta x.$$

$$\boxed{d_{k+1} = d_k + 2\Delta y - 2\Delta x}$$

For initial parameter (P_k)

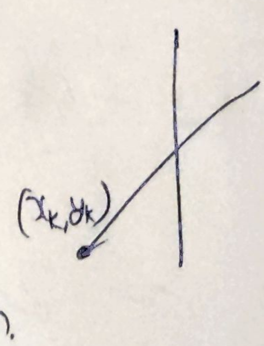
From eqn (iii)

$$\cancel{d_k = 2\Delta y x_k + 2\Delta y - 2\Delta x y_{k+1} + 2\Delta x c - \Delta x}$$

$$d_k = 2\Delta y(x_k + 1) - 2\Delta x y_k + 2\Delta x c - \Delta x$$

Let $y_k = m x_k + c$

$\therefore c = y_k - m x_k.$



Putting the value of c in eqn.

$$d_k = 2\Delta y x_k + 2\Delta y - 2\Delta x y_k + 2\Delta x(y_k - m x_k) - \Delta x$$

$$= 2\Delta y x_k + 2\Delta y - 2\Delta x y_k + 2\Delta x y_k - 2\Delta x y_k$$

$$- 2\Delta x \frac{\Delta y}{\Delta x} x_k - \Delta x.$$

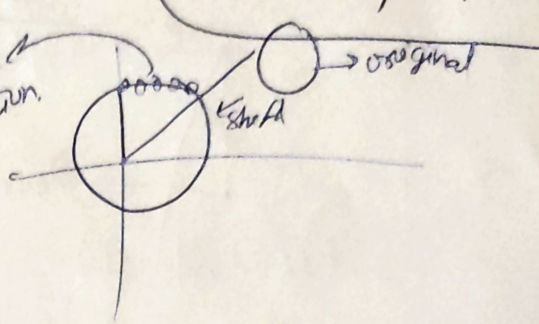
$$\boxed{d_k = 2\Delta y - \Delta x}$$

Final

Circle generation algorithm

Date - 20/03/23

⇒ Bresenham's algo:



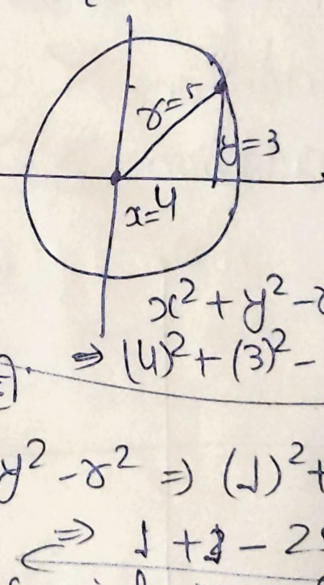
eqn of circle

$$\boxed{x^2 + y^2 = r^2}$$

Or $\boxed{x^2 + y^2 - r^2 = 0.}$

here x, y — any point on the circle.

r = radius eg.



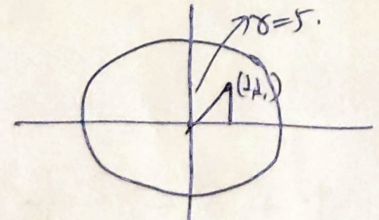
meaning of eqn of circle

0 means the point lies on the circle

eqn of circle में find करते हैं (means) की point circle के अंदर है या नहीं

$x^2 + y^2 - r^2$
$\Rightarrow (4)^2 + (3)^2 - (5)^2 \Rightarrow 0$

eg. (ii)

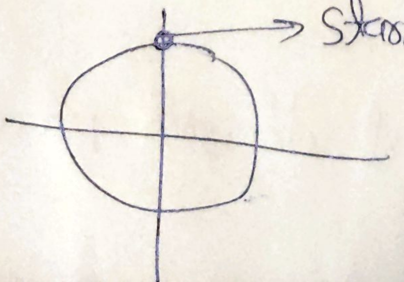


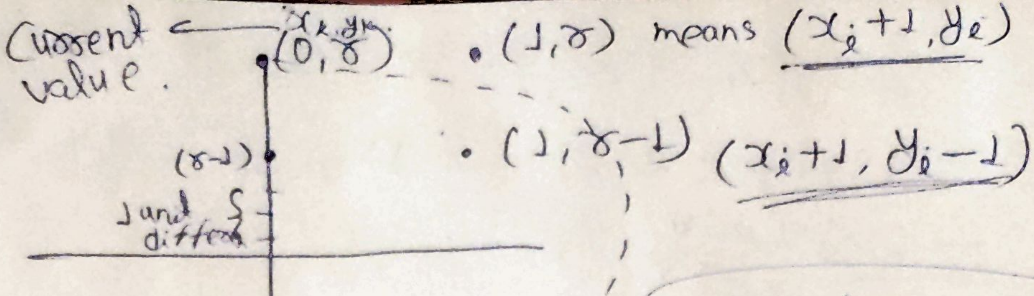
$x^2 + y^2 - r^2 \Rightarrow (1)^2 + (1)^2 - (5)^2$
$\Rightarrow 1 + 1 - 25 \Rightarrow -23 \neq 0$

means point is not in the circle.

Radius (r) = y (height)

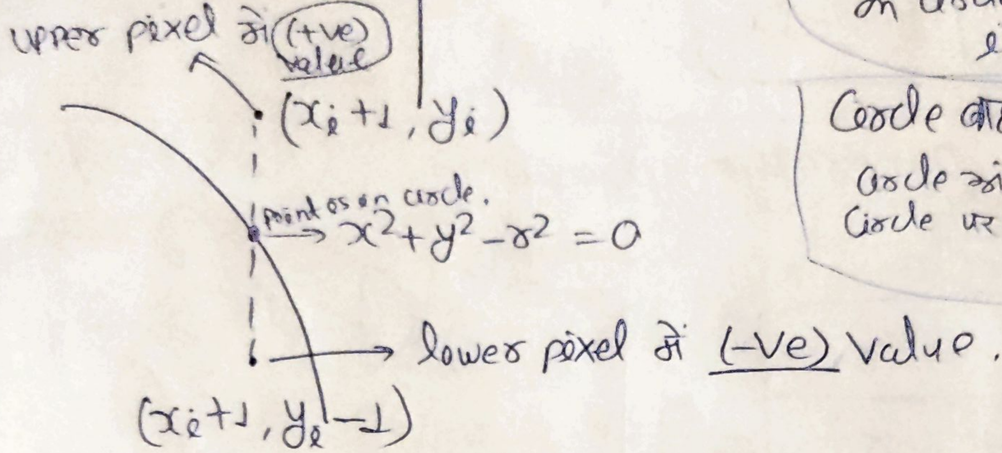
→ Starting point.





In circle x_i is always increase

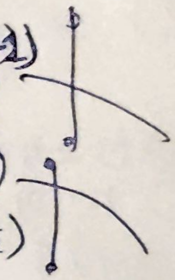
Circle बाहर तो $x^2+y^2-r^2$ तो +ve.
Circle अंदर = -ve
Circle पर = 0



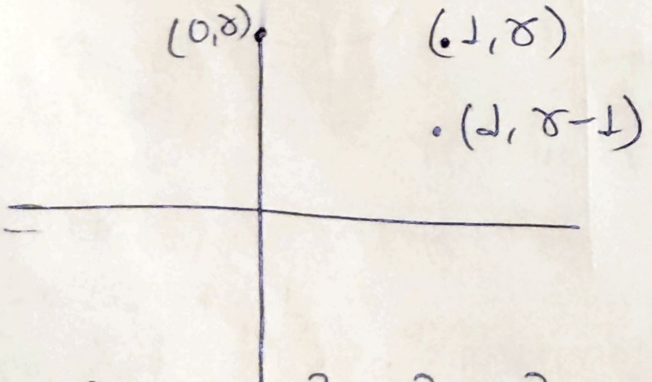
distance का addition.

case ① addition < 0 select ~~lower~~ upper. pixel

case ② addition ≥ 0 select ~~upper~~ pixel lower



$\Rightarrow$



For $(1, r) \Rightarrow x^2+y^2-r^2 \Rightarrow 1^2+r^2-r^2$ — ①

$(1, r-1) \Rightarrow (1)^2+(r-1)^2-r^2$ — (11)

add eqn ① & (11).

$1^2+1+r^2-2r+1-r^2$

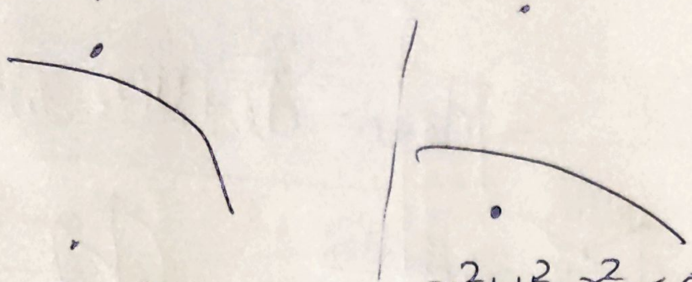
$\Rightarrow \boxed{3-2r}$ $\longrightarrow$ Initial decision parameter eg: If $r = 9$ than initial para. $= 3 - 2r \Rightarrow 3 - 2(9) = -15$

which is $-ve$

means select ~~lower~~ upper pixel (y...

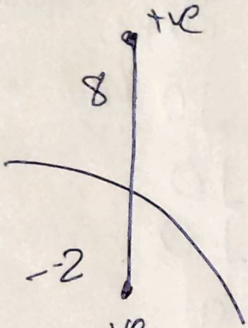
(Date - 21/3/23)

upper | lower



$x^2 + y^2 - r^2 \geq 0$ (+ve) outside of circle | $x^2 + y^2 - r^2 < 0$ (-ve) (inside)

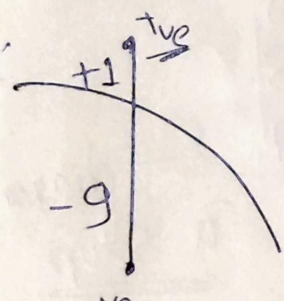
eg.



$8 - 2$ (दोनों को add)

$6 \geq 0$ तो select lower pixel

eg.



$1 - 9 \Rightarrow -8 \Rightarrow -8 < 0$ select upper pixel.

inside circle → -ve

outside circle → +ve

dl और du का addition.

≥ 0 select lower pixel

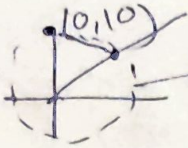
< 0 select upper pixel

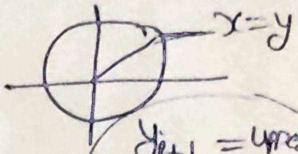
Step (2) Initial parameter $d_i = 3 - 2r$. value < 0 / ≥ 0.
If $r = 10$

(ii) $d_i < 0$, y_i, $x_i + 1$ $\xrightarrow{\text{Next}}$ $\boxed{d_{i+1} = d_i + 4x_i + 6}$

(iii) $d_i \geq 0$, $y_i - 1$, $x_i + 1$ $\longrightarrow$ $\boxed{d_{i+1} = d_i + 4(x_i - y_i) + 10}$

If $r = 10$





y_{i+1} = upper का या निचे का

x_i	y_i	d_i	x_{i+1}	y_{i+1},
0	10	−17 ($3-2r$ For. से)	1	10 (di −ve तो y_{i+1} same)
1	10	−11 ($d_i + 4x_i + 6$ For.)	2	10
2	10	−1	3	10
3	10	13 ($x_i = 2$)	4	9
4	9	−5 ($x_i = 3$)	5	9
5	9	17 ($x_i = 4$)	6	8.
6	8	11 ($x_i = 5$)	7	7.

→ diagonal पे stop

(d_i calculation में upper वाला x_i का value.)

(iii) $-11 + 4(1) + 6 \Rightarrow -11 + 10 \Rightarrow$

(iv) $-1 + 4(2) + 6 \Rightarrow -1 + 8 + 6$

(v) $13 + 4(3 - 10) + 10 \Rightarrow 13 + 4(-7) + 10$
$23 - 28 \Rightarrow -5.$

(vi) $-5 + 4(4) + 6 \Rightarrow -5 + 16 + 6 \Rightarrow 17.$

(vii)

$d_{i+1} = 17 + 4(5-9) + 10.$

$= 17 + 4(-4) + 10$

$= 17 - 16 + 10 \rightarrow 11.$

Stopping condition :-

① when $x_i = y_i$ means $(7 = 7)$

or $x_i > y_i$ or $\underline{8 > 7}$.

Practical

(i) line - generation — 2 algo.

(ii) Circle - generation — 2 algo.

$\boxed{30 \text{ marks}}$

10 w

10 exe.

10 viva.

```
int r;
int _ r
  r=10
  d = 3-2r
if (d<0)
```

1 (marks)

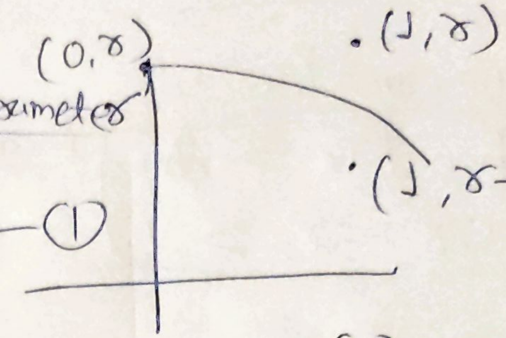
Derivation of all eqn by brehemens algo.

Date - 22/3/23

(1) $3-2r$

(2) $d_i + 4x_i + 6$

(3) $d_i + 4(x_i - y_i) + 10$



2 marks Derive Initial decision parameter

$(1, r) \Rightarrow x^2 + y^2 - r^2$

$\Rightarrow (1)^2 + r^2 - r^2 \Rightarrow 1$ — (I)

$(1, r-1) \Rightarrow (1)^2 + (r-1)^2 - r^2 \Rightarrow 2-2r$ — (II)

$= 1 + r^2 - 2r + 1 - r^2$

$x_{i+1} = x_i$ or x_{i+1}

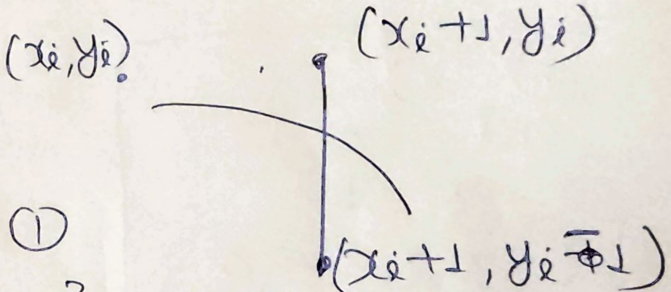
$x_i + 1 = x_i$ हमेशा बढ़ेगा

adding eqn (I) & (II).

$1 + 2 - 2r \Rightarrow \boxed{3-2r}$

& (3)

(2) For :- $d_i + 4x_i + 6$



$(x_{i+1}, y_i) \Rightarrow x^2 + y^2 - r^2$

$\Rightarrow (x_{i+1})^2 + (y_i)^2 - r^2$ — (I)

$(x_{i+1}, y_i - 1) \Rightarrow (x_{i+1})^2 + (y_i - 1)^2 - r^2$ — (II)

adding eqn (I) & (II)

$d_i = (x_{i+1})^2 + (y_i)^2 - r^2 + (x_i+1)^2 + (y_i - 1)^2 - r^2$

$= 2(x_i + 1)^2 + (y_i)^2 + y_i^2 + 1 - 2y_i - 2r^2$

$= 2(x_i + 1)^2 + 2(y_i)^2 - 2y_i - 2r^2 + 1$ — (III)

$i = i + 1$.

$d_{i+1} = 2(x_{i+1} + 1)^2 + 2(y_{i+1})^2 - 2y_{i+1} - 2r^2 + 1$ — (IV)

In next pt $x_{i+1} = x_i + 1$

eqn (iv) − eqn (iii) Put $x_{i+1} = x_i + 1$. 23/3/23

$d_{i+1} - d_i = 2((x_{i+1})+1)^2 + 2(y_{i+1})^2 - 2y_{i+1}$
$\quad -2(x_i+1)^2 - 2(y_i)^2 + 2y_i + \cancel{2}$

In $2((\underline{x_i+1})+1) \rightarrow x_i+1 = a$
$\quad 1 = b.$

$d_{i+1} - d_i = 2\left[(x_i+1)^2 + (1) + 2(x_i+1)\right] + 2((y_{i+1})^2 - (y_i)^2)$
$\quad -2(y_{i+1} - y_i) - 2(x_i+1)^2$

$d_{i+1} - d_i = \cancel{2(x_i+1)^2} + 2 + 4(x_i+1) + 2[(y_{i+1})^2 - (y_i)^2]$
$\quad -2(y_{i+1} - y_i) - \cancel{2(x_i+1)^2}$

$d_{i+1} - d_i = 4x_i + 6 + 2[(y_{i+1})^2 - (y_i)^2]$
$\quad -2[(y_{i+1} - y_i)].$

Condition (i) $\leq 0 \rightarrow$ upper value put $y_{i+1} = y_i$

$d_{i+1} - d_i = 4x_i + 6 + 2[(y_i)^2 - y_i^2] - 2(y_i - y_i)$

$\boxed{d_{i+1} = d_i + 4x_i + 6}$

Condition (ii) $\geq 0 \rightarrow$ lower value, put $y_{i+1} = y_i - 1$

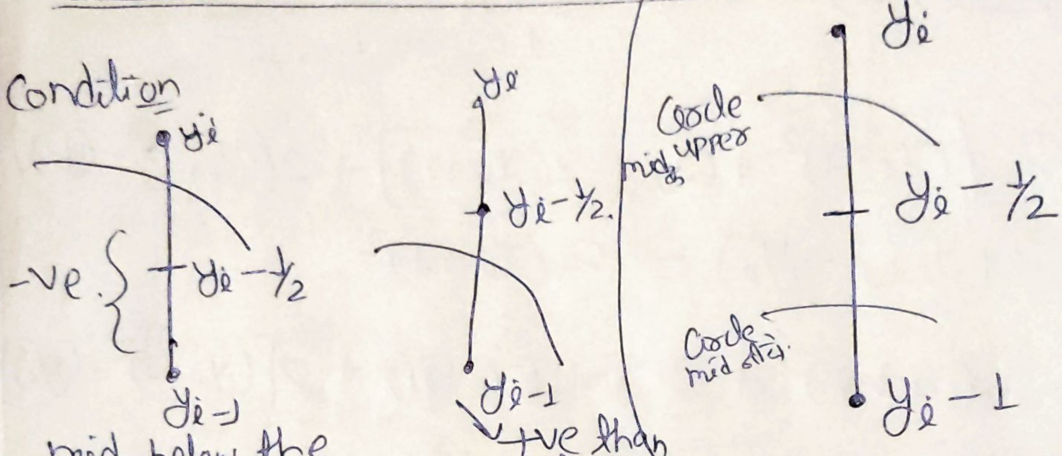
$d_{i+1} - d_i = 4x_i + 6 + 2[(y_i-1)^2 - (y_i)^2] - 2[y_i - 1 - y_i]$
$= 4x_i + 6 + 2[y_i^2 - 2y_i + 1 - y_i^2] - 2[-1]$
$= 4x_i + 6 + 2(-2y_i + 1) + 2.$
$= 4x_i + \cancel{6}8 - 4y_i + 2$

(marks)

Deri:- $d_i = 4x_i - 4y_i + 10$

(1) $\boxed{d_{i+1} = d_i + 4(x_i - y_i) + 10}$

☆ Mid point Circle Generation :-



mid below the circle than it is -ve means upper value.

+ve than select lower pixel.

⇒ Initial parameter

For $M(1, r - \frac{1}{2}) = x^2 + y^2 - r^2$

$\Rightarrow (1)^2 + (r - \frac{1}{2})^2 - r^2$

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

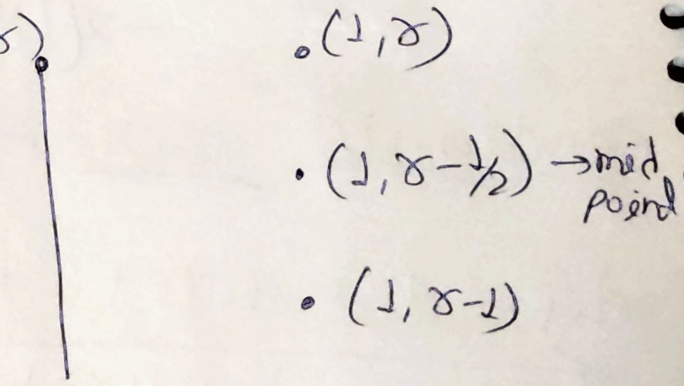
$= 1 - r + \frac{1}{4}$

$\Rightarrow 1 + \frac{1}{4} - r \Rightarrow \boxed{\frac{5}{4} - r}$

skip

or

$= \boxed{1 - r}$



Mid

(Date - 23/3/23)

① $1-r$

② $d_i + (2x_{i+1} + 1) < 0 \Rightarrow$ select (y_i) (x_{i+1} = Next value use)

③ $d_i + 2x_{i+1} - 2y_{i+1} + 1 \geq 0$. select ($y_i - 1$)

If $r = 10$

x_i	y_i	d_i	x_{i+1} (always increase.)	y_{i+1}	$2x_{i+1}$	$2y_{i+1}$
0	10	-9 ($1-r$ for) (−ve)	1	10 (y_i as it is)	2	20
		-6 (−ve)	2	10	4	20
		-1 (−ve)	3	10	6	20
		6 (+ve)	4	9	8	18
		-3 (−ve)	5	9	10	18
		8 (+ve)	6	8	12	16
		5 (+ve)	7	7	14	14

terminating.

$x_{i+1} = y_{i+1}$.

or $x_{i+1} > y_{i+1}$.

(ii) $d_i + 2x_{i+1} + 1$.

$\Rightarrow -9 + 2 + 1 \Rightarrow -6$.

(iii) $-6 + 4 + 1 \Rightarrow -1$

(iv) $-1 + 6 + 1 \Rightarrow 6$

(V) +ve $d_i + 2x_i - 2y_{i+1} + 1$

$\Rightarrow 6 + 8 - 18 + 1 \Rightarrow 7 - 10 \Rightarrow -3$

(vi) −ve $-3 + 10 + 1 \Rightarrow 8$

(vii) +ve

$8 + 12 - 16 + 1$

$9 - 4 \Rightarrow 5$

Derivation of circle by using mid-point :-

① $1 - r$.
② $d_i + 2x_{i+1} + 1$.
③ $d_i + 2(x_{i+1} - y_{i+1}) + 1$.

Formula $d = x^2 + y^2 - r^2$

We have $(x_{i+1}, y_i - 1/2)$ —— (I)

$\therefore d_i = (x_{i+1})^2 + (y_i - \frac{1}{2})^2 - r^2$

$= \cancel{(x_{i+1})^2 + (y_i)^2 - 2 y_i \times \frac{1}{2} + \frac{1}{4}}$

Put $i = i + 1$.

$\therefore d_{i+1} = (x_{i+1} + 1)^2 + (y_{i+1} - 1/2)^2 - r^2$ —— (II).

eqn (II) − eqn (I).

$d_{i+1} - d_i = (x_{i+1} + 1)^2 + (y_{i+1} - 1/2)^2 - \cancel{r^2} - (x_i + 1)^2$
$- (y_i - 1/2)^2 + \cancel{r^2}$.

Put $x_{i+1} = x_i + 1$.

$d_{i+1} - d_i = ((x_i + 1) + (1))^2 + (y_{i+1} - \frac{1}{2})^2 - (x_i + 1)^2 - (y_i - 1/2)^2$

$= \cancel{(x_i + 1)^2} + 2(x_i + 1) + 1 + (y_{i+1})^2 - \cancel{2} \times (y_{i+1}) \times \frac{1}{\cancel{2}}$
$+ \cancel{\frac{1}{4}} - \cancel{(x_i + 1)^2} - (y_i)^2 + \cancel{2} y_i \times \frac{1}{\cancel{2}} - \cancel{\frac{1}{4}}$

$= \cancel{2(x_i) + 2}\ 2(x_i + 1) + (y_{i+1})^2 - y_i^2$
$- (y_{i+1} - y_i) + 1$ —— (III).

Condition ① < 0

Put $y_{i+1} = y_i$.

$$d_{i+1} = d_i + 2(x_i+1) + y_i^2 - y_i^2 - (y_i - y_i) + 1.$$

$$d_{i+1} = d_i + 2(x_i+1) + 1.$$

Put $x_i + 1 = x_{i+1}$ → Next value of x.

$$\boxed{d_{i+1} = d_i + 2x_{i+1} + 1}$$

Condition ⑪ ≥ 0.

Put $y_{i+1} = y_i - 1$.

$$d_{i+1} = d_i + 2(x_i+1) + (y_i - 1)^2 - y_i^2$$
$$- (y_i - 1 - y_i) + 1.$$

$$\cancel{d_{i+1} = d_i + 2(x_i+1) + y_i^2 - y_i^2}$$

$$d_{i+1} = d_i + 2(x_i+1) + \cancel{y_i^2} - 2y_i + 2 - \cancel{y^2}$$
$$+ 1.$$

$$d_{i+1} = d_i + 2(x_i+1) - 2y_i + 2 + 1.$$

$$\cancel{d_{i+1} = d_i + 2[x_{i+1} - y_{i+1}}$$

$$d_{i+1} = d_i + 2(x_i+1) - 2(y_i - 1) + 1.$$

Put $x_i + 1 = x_{i+1}$ & $y_i - 1 = y_{i+1}$.

$$\boxed{d_{i+1} = d_i + 2x_{i+1} - 2y_{i+1} + 1}$$

Random Scan :- based on equation
eg facebook logo , GIF

Raster Scan :- based on pixel. (each pixel information)
eg. • jpeg pic → Photo colours in pixel > 16 million

Random scan at a time 256 colors

256(shades) 256 256
Red • 1 r . b . g
= 0 = 1 = 0

1 pixel = 3 color combin

→ Jpeg compressed

Raster Scan :- Picture quality decrease → data loss.

⇒ How generates > 16 million colors in Raster scan
1 pixel =

Red.	Green	Blue
256	256	256
↓	↓	↓
8 bit	8 bit	8 bit.
0000000 light red.	-//-	-//-
11111111 darkest red.		

R   G   B.

R	G	B
1	1	1
2	2	2
	3	3
	,	4.
	,	,
	,	,
	256	256

Combination $8 \times 8 \times 8 = 24$.

means 2^{24} which is > 16 million.

2 marks Difference b/w Random VS Raster scan:

	Random	Raster
Quality	✓	~~x~~ less.
memory.	~~No~~ less space 256 colors	More space Support 16 M colors
Realistic	X (Unrealistic) eg. cartoon.	✓
Starting point	Start from Random point.	Start from a pixel.

In 1 sec. how many times ~~reps~~ ~~represses~~ refresh on screen →

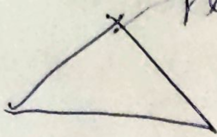
refresh cycle ← 1 sec → 60 times

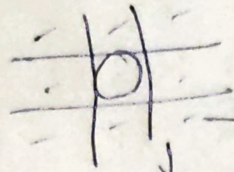
$\boxed{\text{1 sec.} \rightarrow \text{60 times}}$

☆ Polygon filling Algorithm :-

(Date - 27/3/22)

pixel के color.





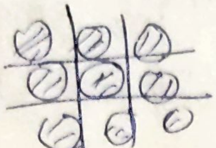
(1,1)

→ 8 connected (all) pixel.

↓ Or 4 connected pixel. ((corner is not imp) (Top-bottom, left & Right))

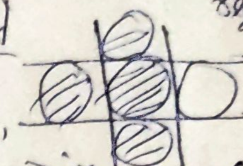
8 = connected

ये 1 भी pixel को खाली नहीं होने देगा।



4 - connected

इसमें कोई ना कोई pixel छूट जाता है,



⇒ 1st random point select जो polygon के अंदर

Step: (A)

⇒ Seed pixel / Starting pixel :-

↳ For finding this their are 2 rule :- (means points अंदर या बाहर check करने के लिए)

(i) Inside - outside rule

(ii) Non - zero winding rule.

(i) Inside - outside rule :-

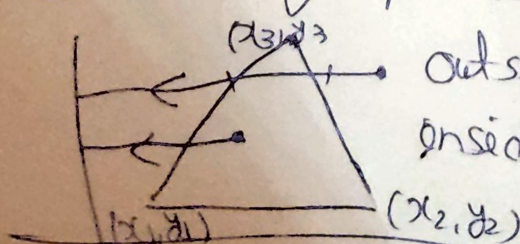
(a) x - axes के parallel करते हुए left / right direction में line.

(b) ~~y = mx+c~~ y → $\boxed{m = \dfrac{y_2 - y_1}{x_2 - x_1}}$ से check करेगा

अगर दोनो slope equal तो

⇒ For: even cutting points ⟶ outside

odd cutting points ⟶ inside.

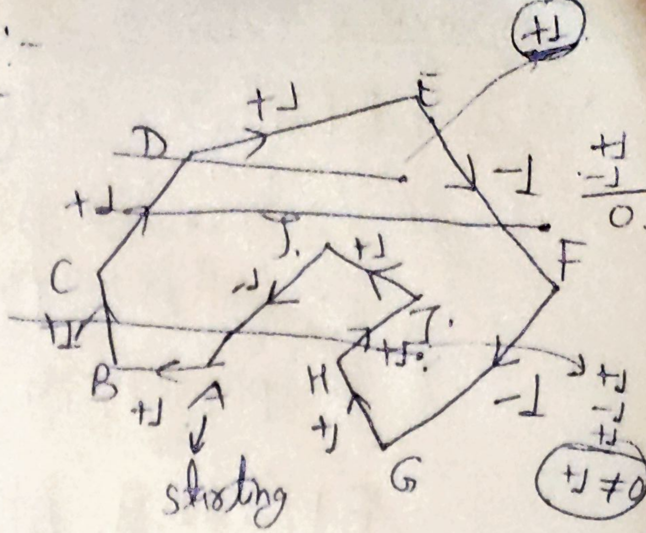


(ii) Non-zero winding rule :-

Concept for choosing one direction

Slope :- (a) increasing

(b) decreasing.



Let we go upper direction

(i) Numbers

> up — increase (+)
>
> down — decrease (-)

(ii) कोई भी point लेकर left / right direction में line.

अगर non-zero ⇒ inside

zero ⇒ outside

fill color.

(बीच वाला कभी भी WHITE नहीं)

(B) How to Color :-

~~8 — Connected pixel~~

Boundary filling algorithm :- सिर्फ 1 color we have.

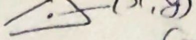
P(x-1, y+1)	P(x, y+1)	P(x+1, y+1)
P(x-1, y)	P(x, y)	P(x+1, y)
P(x-1, y-1)	P(x, y-1)	P(x+1, y-1)

⇒ Process of filling :- Boundary, flood

Checking boundary — boundary वाली & no color is fill

⇒ Boundary filling algo : for 4-connected

pixel colouring



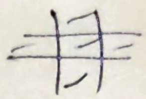
→ for boundary colour

continue হবে যেকু boundary color না মিলে

```
void BF(int x, int y, int fill, int bound)
{ if (getpixel(x,y)! = bound
   && (getpixel(x,y)! = fill) )
  { putpixel(x,y,fill)

     BF(x+1, y, fill, bound)
     BF(x-1, y, fill, bound)
     BF(x, y+1, fill, bound)
     BF(x, y-1, fill, bound)
  }
}.
```

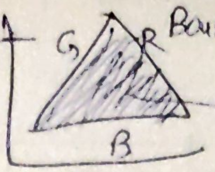
} 4-connected



⇒ 8-connected :- same ~~a~~ 4 connect extra.

Flood filling :- Initially

→ Old color :- it is always unique.



Boundary color कोई सा भी

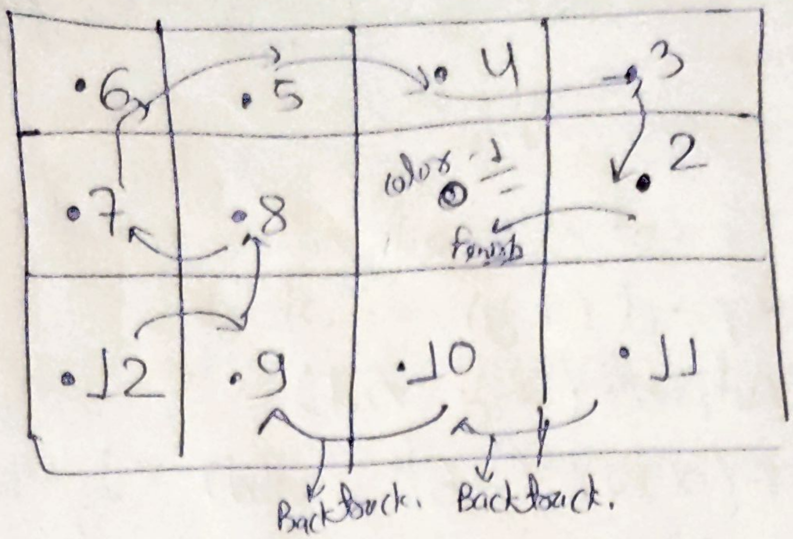
Different color. let it is gray.

let old color → gray , new

```
void FF(int x, int y, int fill, int old)
{ if (getpixel(x,y) == old)
  { putpixel(x,y, fill)
      FF(x+1, y, fill, old)
      FF(x-1, y, fill, old)
      FF(x, y+1, fill, old)
      FF(x, y-1, fill, old)
}}.
```

4-connected

B



4 connected
$x+1, y$
$x+1, y-1$
$x+1, y-1$
$x+1, y-1$

4 connect

(R) $x+1, y$
(L) $x-1, y$.
(Top) $x, y+1$
(Bottom) $x, y-1$.

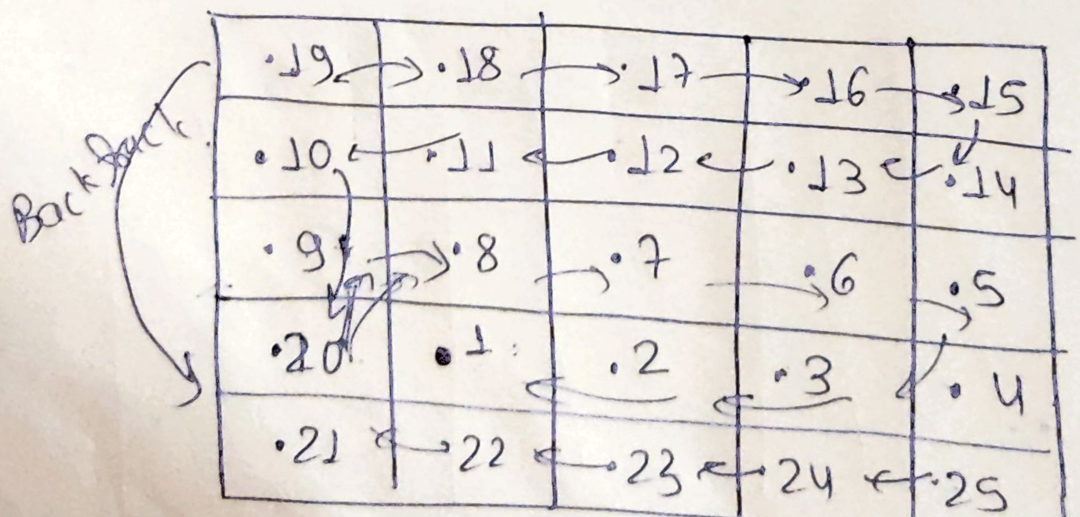
here Boundary filling is (recursive call)

New pixel [illegible] process

1st check → R → L – Top → Bottom

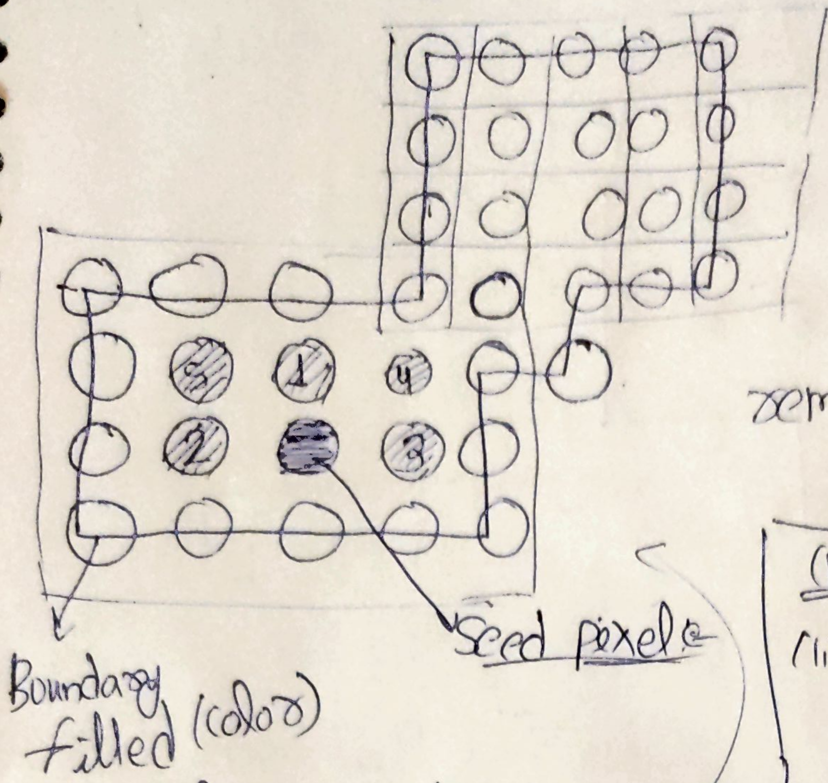
here 11 R→L–Top–Bottom is complete so backtrack to 10.

Eg:- 2-marks.



5X4

Date - 30/3/23



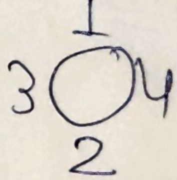
remaining 12 are filled by Stack Method

(i) corresponding seed pixel को ही order
(ii) Next choose pixel which is top of the stack.

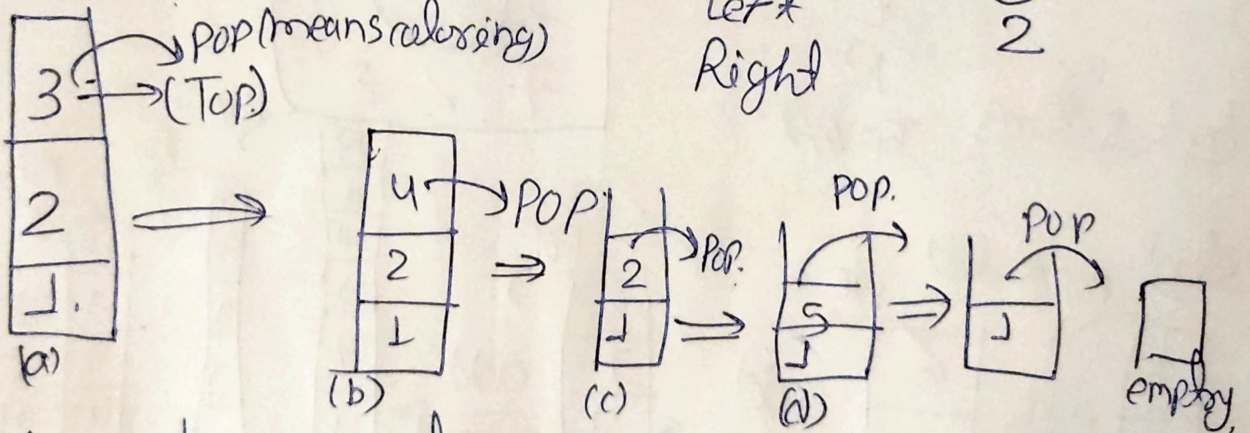
Stack method

→ 4 - connected

order
Top
Bottom
Left
Right



* only check 4-times.



→ coloring when pop out

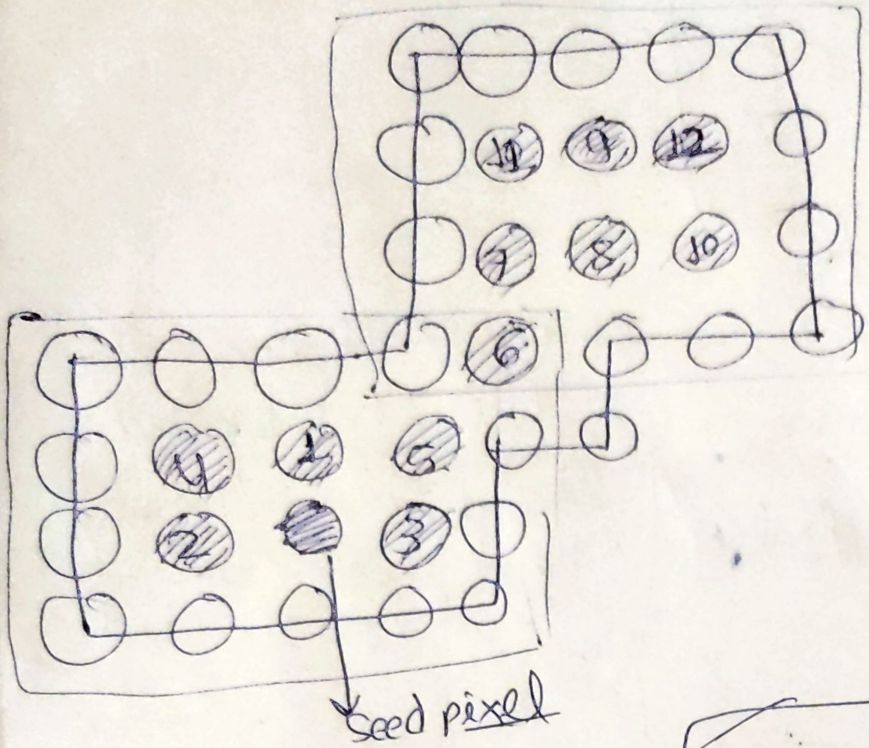
* जिसको POP-out करते हैं, उसको T, B, L, R को Stack में डालना है, जैसे 3 का T, B, L, O.

⇒ In this diagonal are not reached

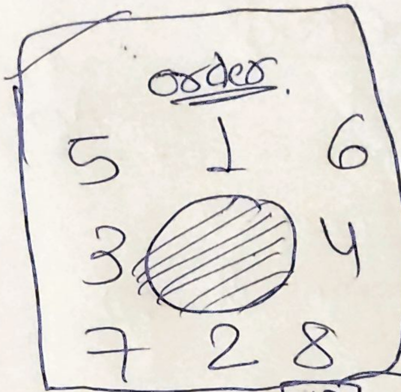
Drawbacks :- cannot process diagonal pixel.

☆ 8-connected

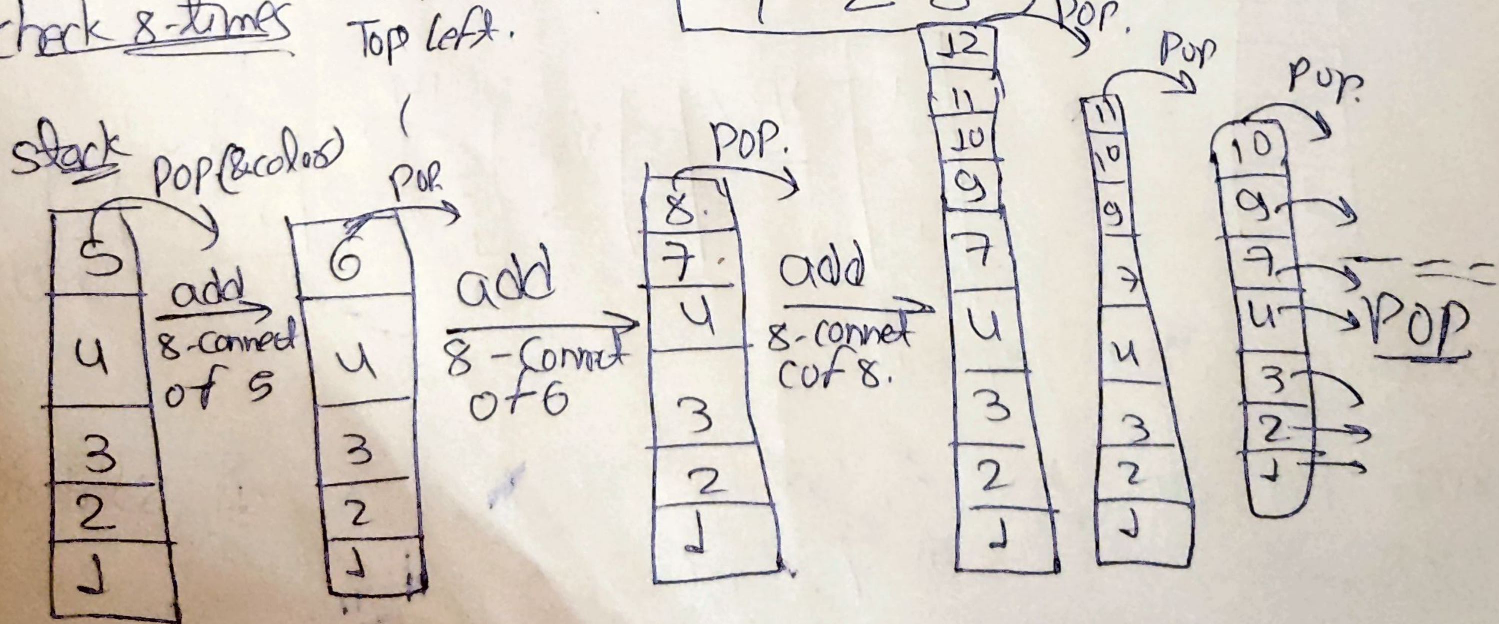
Q. => Diagram से 4-connected & 8-connected का process



order. Top
B
L
R
Top left.



check 8-times



जो color होगा उसके फिर order check