

Syllabus

Lecture - 1

21.1.2026

Introduction

Scan - Conversion

2D Transformation

3D Transformation

Clipping

Polygon filling

Curves



Mid

2D

3D

2D & 3D

mid

clipping

2D & 3D

Final

Introduction:

Computer Graphics कि ?



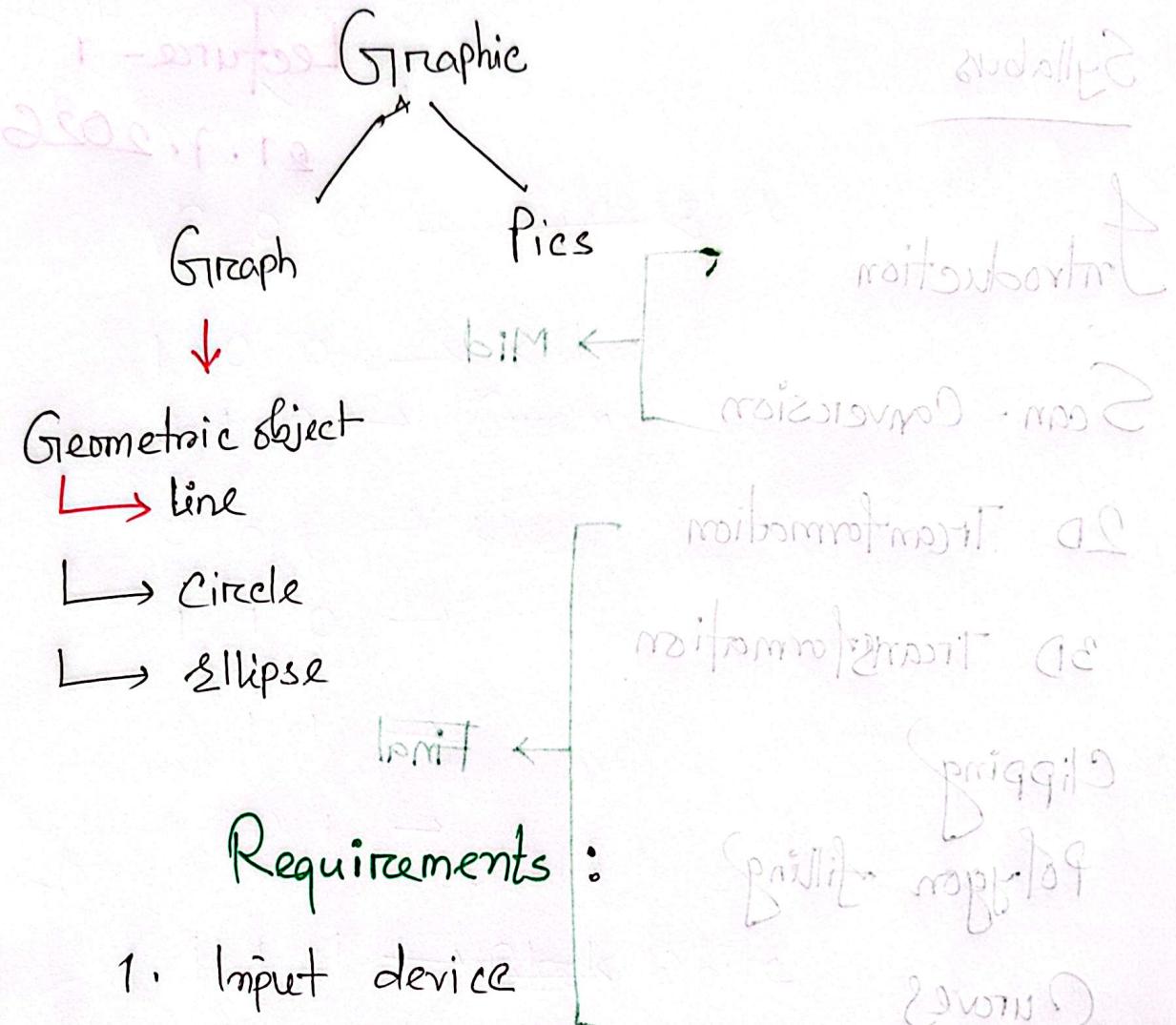
Process of data → Meaningful

meaningful output

Two - 2D (matrix) → Info Generation → Output (1)

Two - 3D → 2D output → Output (2)

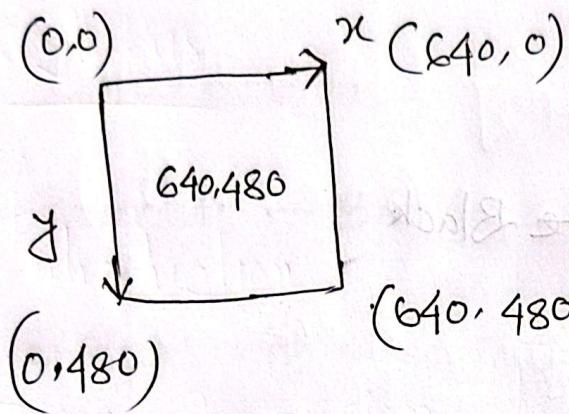
Curves 1/1



Graphics System.

২ প্রকার

- ① Active → এ graphics system output এর user control
করতে পারে আর Active G. S. এর
(Frame)
- ② Passive → Output এর পরে user এর control
- পারে না। (Monitor)



Resolution: ~~প্রতী~~ Maximum number of pixels

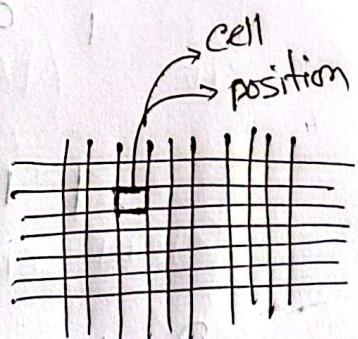
640x480

1024x720 is better cause pixel beshi-

= প্রতিটি cell একেকটি pixel অংশ।

যাই Pixel কর্তৃ মেছি হৈব।

- এন্তু ৬ মেছি হৈব।



Pixel Representation:

দুইটি দিক দেখত হয়

→ Position (x, y)

→ Color

RGB }
CMYK } ৫২ Method

RGB → Display - Q use 128

CMYK → Printer - Q use 255

- এন্ডোডিট Convert ইটে এপ্টে ।

R G B $(0.0.0)^T$ \rightarrow (0.0)

0 0 0 \rightarrow white Black

1 0 0 \rightarrow Red $(0.81.0)^T$

0 1 0 \rightarrow Green

0 0 1 \rightarrow blue

1 1 0 \rightarrow

0 1 1 \rightarrow

1 0 1 \rightarrow

1 1 1 \rightarrow Black White

Pixel value - 3 bit হলুব use করলে \rightarrow 8 bit color
১২৮ পার্থক্য পাওয়া যাবে ।

- 8 bit \rightarrow 256 colors পাওয়া যাবে

24 bit

R G B
8↓ 8↓ 8↓

$$256 \times 256 \times 256 = 16,777,216 = 16.7 \text{ Million}$$

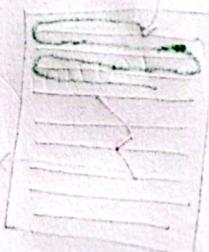
Lecture-2

26/1/26

Version - 17.12

Given version - 3.7.6

Codeblocks with mingw 17.12

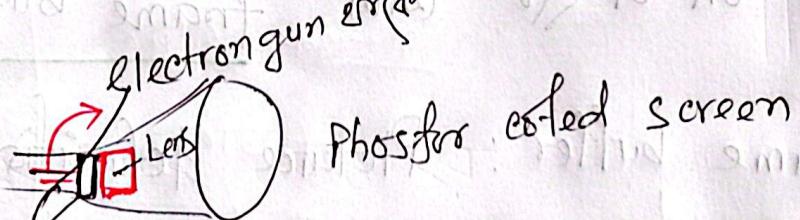


Introduction

2nd slide

Display device

CRT Monitor (Cathode Ray tube)



→ Cathode Ray tube द्वारा तरंग कोन्ट्रोल

→ प्रदूर शक्ति ,

→ Voltage Regulation

→ एचटी इलेक्ट्रॉन मिक्रोवेव फोर्म,

→ Refreshing - २५ (मिनीट) time पर Repeat २५

CRT Monitor

Mechanism

- ① Raster Scan Display (JPEG, GIF, gif)
- ② Vector " " (.doc, .pdf)

Frame buffer ← electron gun visit

① Raster:

ଫାଇଲ୍ ସୁପର୍ଟୋଡ୍

ଏମିହାନ୍ତି କଥା



Row wise divided

ଅଧିକାରୀ ହାତ ନାହିଁ

off line basis

Row wise visit କଥା କଥାନେ ଓ କଥା କଥାନେ

LTR (Left to Right)

(edit and abort) rotation

କଥାନେ ଓ କଥାନେ LTR frame buffer କଥାନେ କଥାନେ

" frame buffer: Picture definition storage

user through command

ଏହା କଥାନେ continuously କଥାନେ ପଦ୍ଧତି କଥାନେ

କଥାନେ

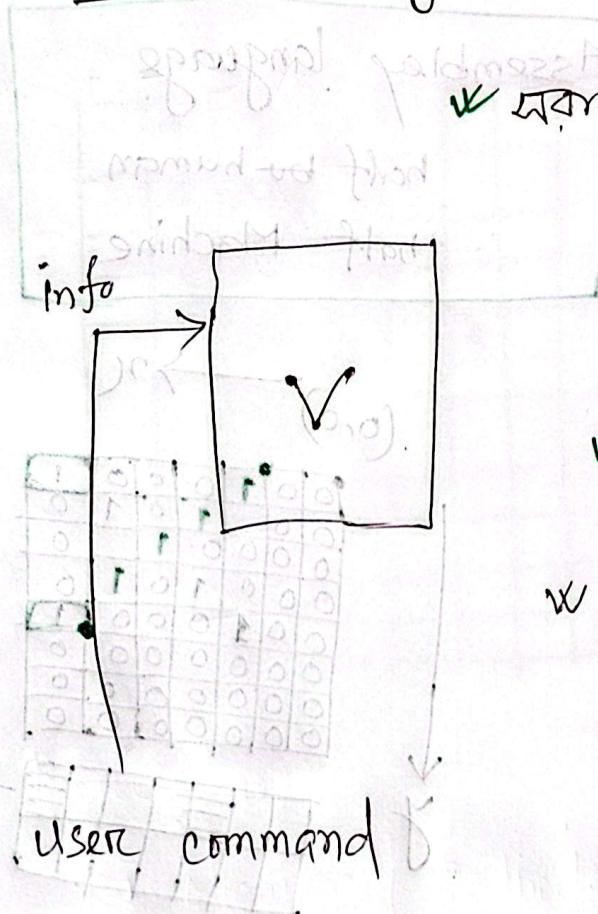
Pixel କଥାନେ କଥାନେ କଥାନେ

Extra hardware, କଥାନେ

Binary

convert କଥାନେ
user com. code

Vector Scan Display



✓ Curve ओर show लेना

✓ complex जटि से

(P-P) वॉल

(P-PP) रास्टर

कार्य सुपोर्ट करता है

SDI + HDI +

■ Frame buffer for 100 frame picture definition

Storage 30-

initialised ← ROM

initialising pattern <-- serial

■ Difference between vector vs Raster

2 User command কিভাবে frame buffer-এ রাখে?



Move, Line
(2,0)

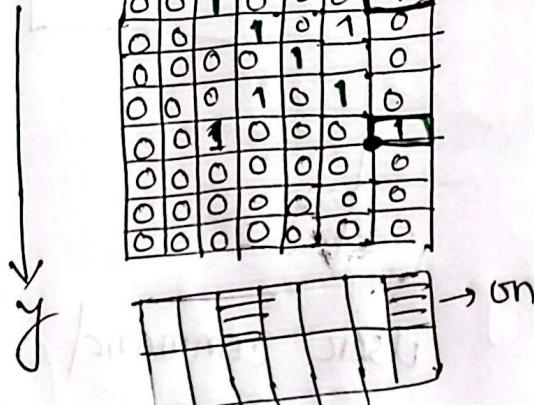
Assembly language.

half by human
half Machine

User Command

Move (2,0)
Line (4,4)
Move (-4,6)
Line (4, -4)

(0,0) → x



x-এর value

+ দান

- মুক্ত

y-এর value

+ নিচে

- ওপর

Line → starting position to
x,y position

Move → Position

↓
Set 1

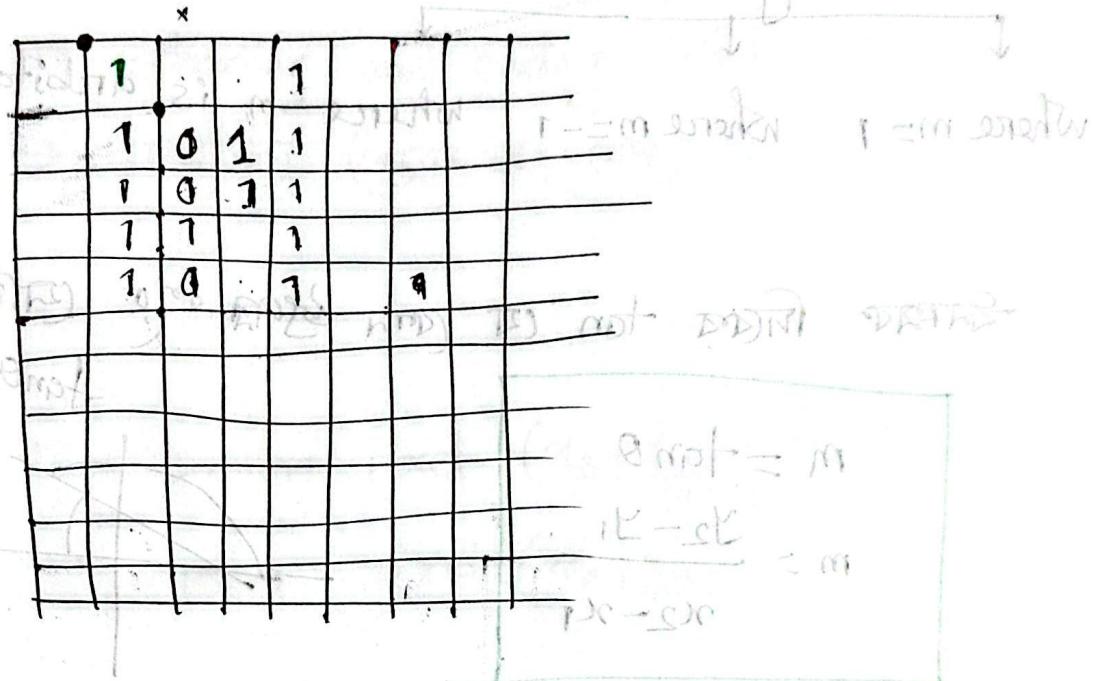
move (1,0)

Line (0,4)

Move (3,0)

Line (0,4)

Line (-3,4)



Lecture 3

2.21-24

Scan Conversion :

Line

circle

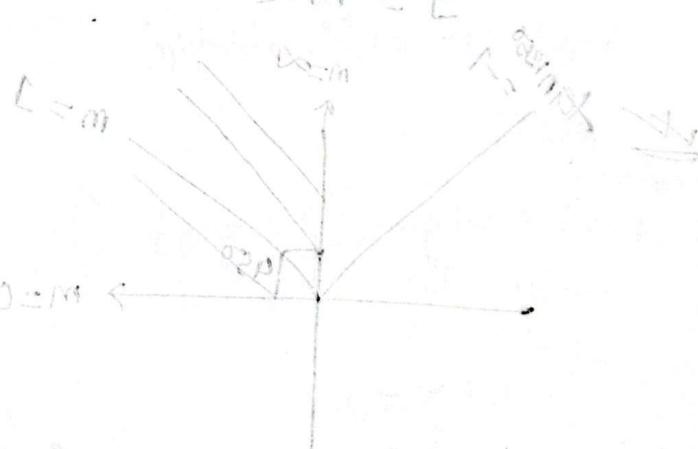
ellipse

Line:

→ Horizontal ($x = \text{const}$)

→ vertical ($y = \text{const}$)

→ Diagonal



Diagonal

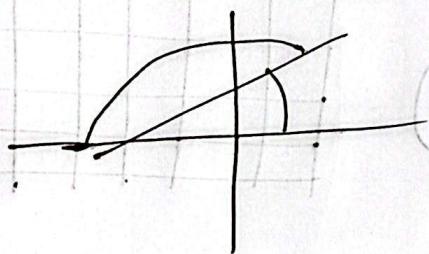
where $m=1$ where $m=-1$ where m is arbitrary

(0,1) sol.

(1,0) sol.

এখানক দিবে $\tan \theta$ যে কোন ক্ষেত্র সম্ভব আছে তার m

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

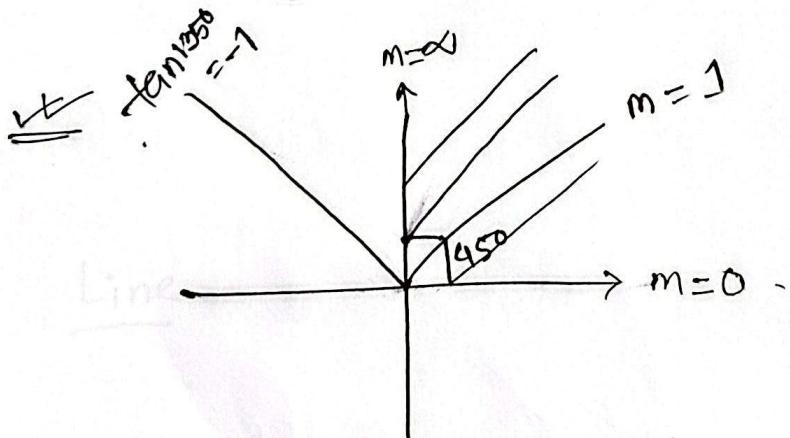


(1,1) sol.

General

$$y = mx + b$$
 (যদিও y এর ফর্ম)

$$y = mx \quad \text{where } b=0$$



General form:

m is arbitrary

m এর যেকোনো মান

($2\pi - x$) ক্ষেত্রে অসম্ভব

($\pi/2$) ক্ষেত্রে অসম্ভব

লাগভী

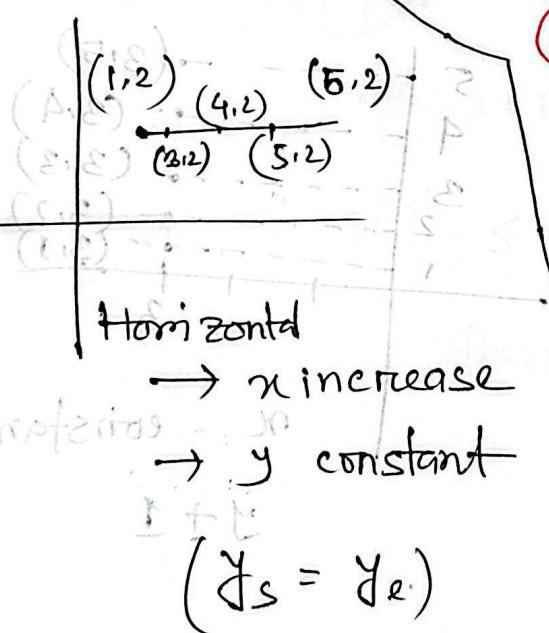
Horizontal Line

Line এর প্রথম starting and ending pixel
end point → প্রান্তবিন্দু (ক্রমে শুরু শেষ)

Algorithm

Pseudocode

1. Input + starting pixel (x_s, y_s)
 ending pixel (x_e, y_e)



2. if $(x_s > x_e)$ then swap
 $(x_s \leftrightarrow x_e)$

3. Initialization $x=x_s$,
 $y=y_s$

4. Loop : Setpixel (x, y)
with color
 $x = x + 1$ → Graphics
if $(x < x_e) \text{ go to loop}$
otherwise exit

Float value নিতে হবে,

and 1 bitness both

Glut \rightarrow $x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_6 \wedge x_7 \wedge x_8 \wedge x_9 \wedge x_{10} \wedge x_{11}$

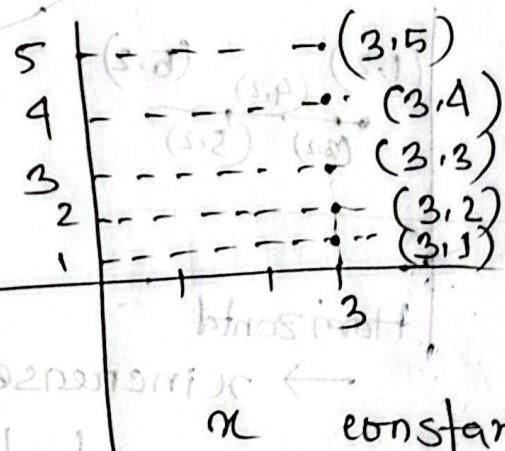
মনের (language) use কোন ওর hedarr

Lecture 4

(28 * 20) bitig primate + frag! 3, 02, 2026

(28 * 20) bitig gibba

Vertical Line:



x constant

$y+1$

$$(2B = 2B)$$

(80c) bitig + 2 good
order diff

$$2L = L$$

$$L + 2c = 10$$

$$^oB(x > 2c) + 1$$

good or of

fixe seforce

L

Algorithm:

1. Input

starting pixel (x_s, y_s) and ending pixel (x_e, y_e)

2. If $(y_s > y_e)$, then swap $y_s \leftrightarrow y_e$

3. Initialization $x = x_s, y = y_s$

4. Loop: Set pixel (x, y) with color

$y = y + 1$

if $(y \leq y_e)$ go to loop

otherwise exit

$$\frac{B+D}{20D} = m$$

④ Easier to do now

⑤ Easier to do

works with lines

works with polygons

Diagonal Line where $m=1$

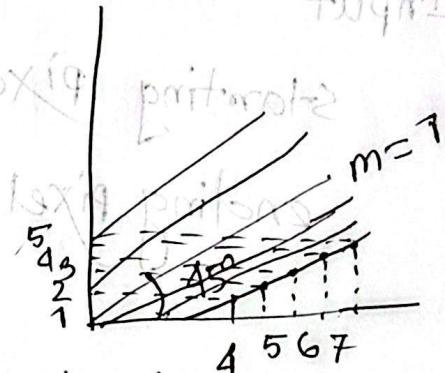
Algorithm

Algorithm:

1. Input

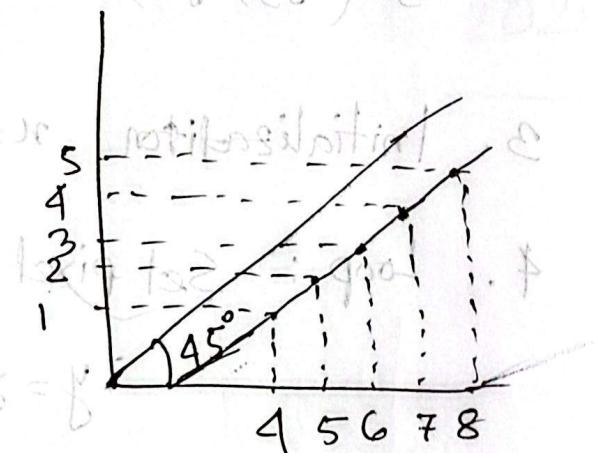
starting pixel (x_s, y_s)

Ending pixel (x_e, y_e)



2. If $(x_s > x_e)$ then

swap $x_s \leftrightarrow x_e, y_s \leftrightarrow y_e$



3. Initialization $x=x_s, y=y_s$

4. Loop: Setpixel (x, y) with color
($4,1$) ($5,2$) ($6,3$) ($7,4$)

$x = x + 1$

$y = y + 1$

if $(x \leq x_e)$ go to loop

otherwise exit

✓ x এর একটি ক্রম যাইছে

✓ x এর y এর difference
একই

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

✓ x এর একটি যাইছে

বিন্দু যাইছে

✓ condition always
independent variable.

Diagonal Line where $m = -1$

Algorithms:

1. Input →

Starting pixel (x_s, y_s)

Ending pixel (x_e, y_e)

2. If $(x_s > x_e)$ then swap $x_s \leftrightarrow x_e$,

$y_s \leftrightarrow y_e$

3. Initialization $x = x_s, y = y_s$

4. Loop : Setpixel (x, y) with color

$x = x + 1$

$y = y - 1$

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

$$\Delta y = -\Delta x$$

if $(x < x_e)$ go to loop

otherwise exit

$$\Delta x = 1$$

Diagonal Line where m is arbitrary

for types

① Direct Line Drawing Algorithm

② DDA (Digital Differential Analyzer) Algorithm

③ Bresenham's Algorithm

प्रायः Line एवं equation

$$(y = mx + c)$$

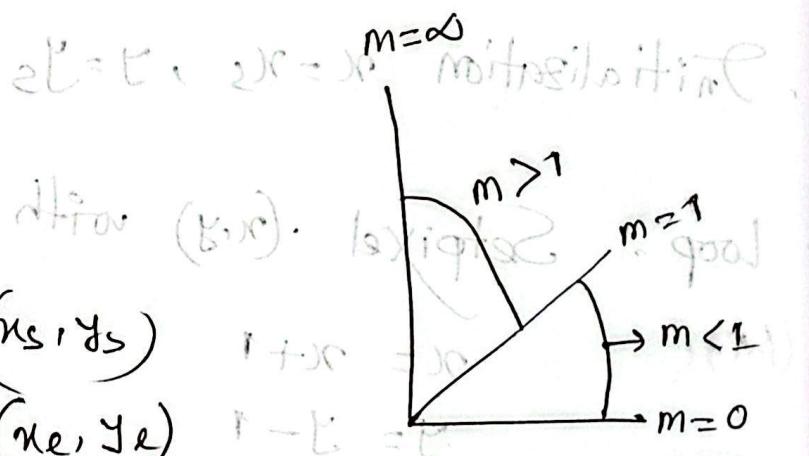
① Direct Line drawing Algo

Algorithm

1. Input

→ Starting pixel (x_s, y_s)

Ending pixel (x_e, y_e)



2. If $(x_s > x_e)$, then swap

$x_s \leftrightarrow x_e, y_s \leftrightarrow y_e$

3. Initialization $x = x_s, y = y_s$

$m++$ (increase)
y calculation
एकेत चर्चा

$$y = mx + b$$

$$4. m = \frac{(y_c - y_s)}{(x_c - x_s)}$$

$\rightarrow [m \text{ এর মান } (যে কোনো নিয়েই)$

$$5. b = y_s - mx_s$$

(1) b এর মান
(2, 3) পরিবর্ত

6. Loop : Setpixel (x, y) with colors

$$x = x + 1$$

$$y = mx + b, \text{ Round}(y)$$

if ($x < x_e$) go to loop.

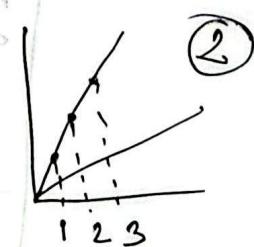
otherwise 'exit'

এই Algorithm efficient নি

Disadvantage:

1. Computation-time high

2. $m > 1$ এর জন্য অনেক মাত্রা লেখা হবে না গড়-গড়(প্রস্তুতি) গ্রাফ কোর্টে,



তারপর

DDA Algorithm

সহজ

Maths

$$\frac{(2^k - 3^k)}{(2^k - 3^k)} = m$$

Starting (1, 1)

Ending (4, 5)

$$2^{2019} - 3^k = 0 \rightarrow k$$

④

$$m = \frac{(5-1)}{(4-1)} = \frac{4}{3}$$

⑤

$$b = 1 - \frac{4}{3} \cdot 1$$

$$b = -\frac{1}{3}$$

or $y = \frac{4}{3}x - \frac{1}{3}$

Answers

x	$y = mx+b$	Round(y)	(x, y)
1	1	1	(1, 1)
2	$\frac{4}{3} \cdot 2 - \frac{1}{3}$ $= \frac{7}{3} = 2.33$	2	(2, 2)
3	$\frac{4}{3} \cdot 3 - \frac{1}{3}$ $= \frac{11}{3} = 3.66$	4	(3, 4)
4	$\frac{4}{3} \cdot 4 - \frac{1}{3}$ $= \frac{15}{3} = 5$	5	(4, 5)

Calculation

ପ୍ରାଚୀର କିମ୍

ନେଟ୍‌କୋମ୍

ଲୋଟି

