CG & MA Notes by Bhavy Sharma

Q1. Give the application of Multimedia?
Ans. Multimedia is used in various fields due to its integration of text, images, audio, video, and animations. Some key applications include:

1. **Education** – Used in e-learning, digital classrooms, and interactive training modules.
2. **Entertainment** – Found in video games, movies, music streaming, and animations.
3. **Business & Marketing** – Used in advertisements, product demos, and corporate presentations.
4. **Healthcare** – Applied in medical imaging, surgery simulations, and virtual consultations.
5. **E-Commerce** – Enhances online shopping with interactive catalogs and virtual try-ons.
6. **Social Media & Communication** – Used in video conferencing, social media platforms, and vlogs.

Q2. Define Pivot Point rotation of object?
Ans. **Pivot Point Rotation of an Object**

Pivot Point Rotation is a transformation technique in computer graphics where an object is rotated around a fixed point called the **pivot point**. The pivot point can be at the **center** of the object, a **corner**, or any **arbitrary position**.

Rotation Formula:

For a point **(x, y)** rotated by an angle **θ** around the origin:

$$x' = x\cos\theta - y\sin\theta$$

$$y' = x\sin\theta + y\cos\theta$$

Q3. Define output Primitive?
Ans.

Output Primitives

Output Primitives are the basic building blocks used in computer graphics to create images and shapes on a display screen. These primitives include fundamental graphical elements such as **points, lines, circles, polygons, and text**.

Common Output Primitives:

1. **Point** – A single pixel on the screen.
2. **Line** – A straight connection between two points.
3. **Circle & Ellipse** – Defined using mathematical equations.
4. **Polygons** – Closed shapes like triangles, rectangles, etc.
5. **Text** – Displaying characters and symbols.

Q4. Define Aspect ratio?
Ans. **Aspect Ratio**

Aspect Ratio is the proportional relationship between the width and height of an image, screen, or display. It is defined as:

$$\text{Aspect Ratio} = \frac{\text{Width}}{\text{Height}}$$

Q5. Define Scan Conversion?
Ans. **Scan Conversion**

Scan Conversion is the process of converting **mathematical representations** of geometric shapes (such as lines, circles, and polygons) into **pixel-based images** for display on a screen. This process is essential in **raster graphics**.

Key Aspects of Scan Conversion:

1. **Line Drawing Algorithms** (e.g., DDA, Bresenham's Algorithm).
2. **Circle Drawing Algorithms** (e.g., Midpoint Circle Algorithm).
3. **Polygon Filling** (e.g., Flood Fill, Scanline Algorithm).

Q6. Define convex and concave Polygon?

Ans. Convex and Concave Polygons

A **polygon** is a closed shape with multiple sides. Based on the arrangement of its vertices and angles, polygons are classified as **convex** or **concave**.

1. Convex Polygon

- A polygon in which **all interior angles are less than 180°**.
- A line segment joining any two points inside the polygon **always lies within** the polygon.
- Example: Triangle, Regular Hexagon.

2. Concave Polygon

- A polygon in which **at least one interior angle is greater than 180°**.
- A line segment joining two points inside the polygon **may pass outside** the polygon.
- Example: Star Shape, Arrowhead.

Q6. Define Application of CG**?

Ans. Applications of Computer Graphics (CG)

Computer Graphics (CG) is widely used in various fields for creating and manipulating visual content. Some key applications include:

1. Entertainment & Animation 🎥🎮

- Used in movies for visual effects (VFX) and animations.
- Video games with realistic 3D environments.
- Virtual reality (VR) and augmented reality (AR).

2. CAD (Computer-Aided Design) 🏗️

- Designing buildings, cars, and machinery.
- Architectural visualization and 3D modeling.
- Used in mechanical and civil engineering.

3. Scientific Visualization 🔬

- Representing complex scientific data graphically.
- Medical imaging (CT scans, MRIs, and 3D reconstructions).
- Weather forecasting and climate modeling.

Q7. Define the concept of Vanishing Point?

Ans. Concept of Vanishing Point

it's the point on a 2D image where parallel lines, extending into the distance in a 3D scene, appear to meet, creating the illusion of depth and perspective.

Features:

- Used in **perspective drawing** to create realistic scenes.
- In **one-point perspective**, all lines meet at a single vanishing point.
- In a two-point **or three-point perspective**, multiple vanishing points exist.

Q8. Define Refresh/Frame Buffer?

Ans. Refresh Buffer / Frame Buffer

A **Refresh Buffer** or **Frame Buffer** is a dedicated memory area in a computer graphics system that **stores pixel data** for an image before it is displayed on the screen.

Key Features:

- It holds the **intensity and color** values of each pixel.
- The stored image is **continuously refreshed** on the screen to maintain a stable display.
- Used in **raster scan displays** for rendering graphics.

Types of Buffers in Computer Graphics:

1. **Color Buffer** – Stores color values of pixels.
2. **Depth (Z) Buffer** – Stores depth information for 3D rendering.
3. **Stencil Buffer** – Used for masking and special effects.

Q9. Define video display devices?

Ans. Video Display Devices

Video Display Devices are hardware components used to visually present digital information, images, and graphics. They play a crucial role in computer graphics and multimedia applications.

Types of Video Display Devices:

1. **Liquid Crystal Display (LCD)**
    - Uses liquid crystals to control light and display images.
    - Commonly used in laptops, TVs, and mobile screens.
2. **Light Emitting Diode (LED) Display**
    - An advanced version of LCD with better brightness and energy efficiency.

- Used in modern monitors, TVs, and billboards.

3. **Plasma Display**

    - Uses small gas cells (plasma) to create high-contrast images.
    - Mostly found in large television screens.

4. **Organic LED (OLED) Display**

    - Uses organic compounds that emit light for better contrast and flexibility.
    - Used in high-end smartphones, TVs, and gaming devices.

Q10. Define the drawback of DDA line generation Algo & Advantage of Bresenhan's line Algo?

Ans. ## Drawbacks of DDA Line Generation Algorithm

The **Digital Differential Analyzer (DDA) Algorithm** is a simple line-drawing algorithm but has some drawbacks:

1. **Floating-Point Computation** – Uses real-number calculations, which are slower and less efficient.

2. **Rounding Errors** – Continuous floating-point operations can introduce rounding errors, causing small inaccuracies.

3. **More Computation** – Requires multiple additions and multiplications, making it slower for real-time applications.

Advantages of Bresenham's Line Algorithm

The **Bresenham's Line Algorithm** is an improved method for drawing lines, offering several advantages:

1. **Uses Only Integer Arithmetic** – It avoids floating-point calculations, making it faster.

2. **More Accurate** – It minimizes rounding errors by using a decision variable.

3. **Efficient for Hardware Implementation** – Can be implemented in graphics hardware for real-time rendering.

4. **Better Performance** – Works efficiently for shallow and steep lines.

Q11. What is the Animation File Format? List some Animation Software.

Ans. ## Animation File Format

An **Animation File Format** is a digital file type used to store animated images, sequences, and motion graphics. These formats contain multiple frames that create motion when played in sequence.

Common Animation File Formats:

1. **GIF (Graphics Interchange Format)** – Supports short animations and transparency.

2. **SWF (Shockwave Flash)** – Used for web-based animations (Adobe Flash).

3. **MP4 (MPEG-4 Video Format)** – Common for high-quality animated videos.

4. **AVI (Audio Video Interleave)** – Supports high-quality animations with audio.

5. **FLV (Flash Video Format)** – Used for streaming animated content.

6. **MOV (Apple QuickTime Movie)** – High-quality animation format used in media production.

List of Animation Software

1. **Adobe Animate** – Formerly Flash, used for 2D animation.
2. **Autodesk Maya** – Industry-standard 3D animation software.
3. **Blender** – Open-source 3D modeling and animation tool.
4. **Toon Boom Harmony** – Used for professional 2D animation.
5. **Synfig Studio** – Free and open-source 2D animation software.
6. **Pencil2D** – Simple software for hand-drawn animation.
7. **Cinema 4D** – Popular for motion graphics and 3D animation.
8. **Moho (Anime Studio)** – Used for vector-based 2D animation.

Q12. Compare Window part & View Port.
Ans.

Aspect	Window	Viewport
Definition	Selected area of the scene in world coordinates.	Area on screen where the scene is displayed.
Coordinate System	World coordinate system (logical units).	Device coordinate system (pixels).
Purpose	Defines what to display.	Defines where to display.
Role	Source rectangle.	Destination rectangle.
Transformation	Acts as input for transformation.	Acts as output/display result.

Q13. Explain the following composite transformations.

- **Translation**
- **Rotation**

Ans. Bhavy

1. Translation

Translation is a transformation that **moves an object from one location to another** in the coordinate system without changing its shape, size, or orientation.

Mathematical Formula:
If a point P(x,y) is translated by T(tx,ty), then the new coordinates become:

$$P'(x', y') = (x + tx,\ y + ty)$$

2. Rotation

Rotation is a transformation that **rotates an object** about a fixed point (usually the origin) by a specific angle θ, either in a clockwise or counter-clockwise direction.

Mathematical Formula (about origin):

$$x' = x \cdot \cos\theta - y \cdot \sin\theta$$

$$y' = x \cdot \sin\theta + y \cdot \cos\theta$$

Q14. Explain Multimedia: Importance of Animation in Multimedia.

Ans. What is Multimedia?

Multimedia is the integration of multiple forms of media such as **text, images, audio, video, graphics, and animation** to convey information in an interactive and engaging way.
It is widely used in education, entertainment, advertising, and web applications to make communication more effective and appealing.

---

2. Importance of Animation in Multimedia

Animation plays a **crucial role** in multimedia applications due to the following reasons:

a) Enhances Visual Appeal

- Animation makes content more attractive and engaging.
- It helps in grabbing users' attention effectively.

b) Improves Understanding

- Complex concepts can be simplified and explained better through animated visuals.
- Especially useful in **e-learning** and **technical education**.

c) Adds Interactivity

- Animations in buttons, menus, and transitions improve user experience in apps and websites.

d) Better Communication

- Ideas and messages are communicated more clearly and dynamically through motion graphics and animated storytelling.

e) Effective Marketing

- Animated advertisements and explainer videos are widely used in digital marketing to attract customers.

f) Simulation and Training

- Used in flight simulators, medical training, and industrial training to mimic real-world scenarios.

Q15. Write short note on.

- **Cyrus - Beck Algo**
- **Bezier Curve**

Ans. (a) Cyrus-Beck Algorithm

The **Cyrus-Beck Algorithm** is a **line clipping algorithm** used for **convex polygonal clipping windows**, Cyrus-Beck can clip lines against any convex polygon.

Key Points:

- Based on **parametric representation** of a line segment.
- Works by calculating the value of parameter t for entering and exiting points.
- It checks the **dot product** of normal vector and direction vector to determine visibility.
- Efficient and accurate for convex shapes.

(b) Bézier Curve

A **Bézier Curve** is a **parametric curve** used in computer graphics to model smooth and scalable curves.

- Defined using **control points**:
    - **Linear Bézier**: 2 points
    - **Quadratic Bézier**: 3 points
    - **Cubic Bézier**: 4 points (most commonly used)

- The shape of the curve is influenced by the position of control points.
- **Mathematically based on Bernstein polynomials**.
- Used in animation, font design, vector drawing, etc.

Q15. What is Clipping? Explain the Midpoint subdivision algo for Line Clipping.
Ans. Bhavy

What is Clipping?

Clipping is the process of **removing parts of a graphic object** (like a line, polygon, or image) that lie **outside a specified region**, called the **clipping window**.
It helps in displaying only the visible portion of an object and discarding the rest.

🔹 **Used for:**

- Viewport display
- Object rendering
- GUI applications
- Game development

---

2. Midpoint Subdivision Algorithm (for Line Clipping)

The **Midpoint Subdivision Algorithm** is a **recursive line clipping technique** used to clip lines against a **rectangular clipping window**.

Advantages:

- Simple and easy to implement.
- Good for lines that cross the clipping boundary.
- More accurate than some other algorithms for fine clipping.

Disadvantages:

- **Slow** for long or complex lines due to repeated subdivisions.
- Less efficient than **Cohen-Sutherland** for performance-critical applications.

(Out of Syllabus)

Date ___ **Repeat in 2012,2013,2018**

Q.1 Z-Buffer method or Depth-Buffer Method

Sol^n- When viewing a picture containing non transparent objects and surfaces, it is not possible to see those objects from view which are behind from the objects closer to eyes. To get the realistic screen image, removal of these hidden surfaces is must. The identification and removal of these surfaces is called as the Hidden-surface problem.

Z-buffer method is one of the commonly used method for hidden surface detection. It is an Image-space method which is based on the pixel to be drawn on 2D.

The Z-buffer method compares surface depths at each pixel position on the projection plane. Normally z-axis is represented as the depth. The algorithm for Z-buffer method is given below:

Step 1:- First of all, initialize the depth of each pixel.
i.e, $d(i,j)$ = infinite (max length)

Step 2:- Initialize the color value for each pixel as $c(i,j)$ = background color.

Step 3:- For each polygon, do the following steps:

Date ___/___/_____ Page No.: _____

```
for (each pixel in polygon's projection)
{
    find depth i.e., z of polygon
    at (x,y) corresponding to pixel(i,j)
        if (z < d(i,j))
        {
            d(i,j) = z;
            c(i,j) = color;
        }
}
```

Points to rember:

1. Z buffer method does not require pre-sorting of polygons.
2. This method can be executed quickly even with many polygons.
3. This can be implemented in hardware to overcome the speed problem.
4. No object to object comparison is required.
5. This method can be applied to non-polygonal objects.
6. Hardware implementation of this algorithm are available in some graphics workstations.
7. The method is simple to use and does not require additional data structure.
8. The z-value of a polygon can be calculated incremetally.

Ques. Perform a 60° rotation of Triangle A(0,0), B(1,1) and C(-1,-1):
(a) about origin
(b) about P(0,1).

2012- June

Solⁿ:- (a) First, let's find the co-ordinates of each point after rotation about origin —

For Point A(0,0):

$$A' = \begin{bmatrix} \cos 60^\circ & -\sin 60^\circ \\ \sin 60^\circ & \cos 60^\circ \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

For Point B(1,1):

$$B' = \begin{bmatrix} \cos 60^\circ & -\sin 60^\circ \\ \sin 60^\circ & \cos 60^\circ \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} - \frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} + \frac{1}{2} \end{bmatrix}$$

For Point C(-1,1):

$$C' = \begin{bmatrix} \cos 60^\circ & -\sin 60^\circ \\ \sin 60^\circ & \cos 60^\circ \end{bmatrix} * \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} - \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} - \frac{1}{2} \end{bmatrix}$$

(b) Now, let's perform a 60-degree rotation about point P(0,1). To do this, we'll first translate the triangle so that P becomes the origin, perform the rotation, and then translate the triangle back!

Translate Triangle ABC so that point P becomes the origin!

$A''(0-0,\ 0-1) = (0,-1)$

$B''(1-0,\ 1-1) = (1,0)$

$C''(-1-0,\ 1-1) = (-1,0)$

Now perform the rotation about the origin using the same rotation matrix!

$$\begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix} * \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} -\frac{\sqrt{3}}{2} \\ -\frac{1}{2} \end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix} * \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix} * \begin{bmatrix} -1 \\ 0 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix}$$

Translate the triangle back to its original position:

$A'''(0+0, -1+1) = (0,0)$

$B'''(1+0, 0+1) = (1,1)$

$C'''(-1+0, 0+1) = (-1,1)$

2012

Ques. Explain Painter's Algorithm.

Sol^n: The painter's algorithm is a simple algorithm used in computer graphics to determine the visibility of surfaces or objects in a scene when rendering a 3D scene into a 2D image. It's named after the analogy of a painter who starts painting the scene from the back and moves towards the front, painting each object in turn, covering the previously drawn objects.

Here's how the Painter's algorithm works:

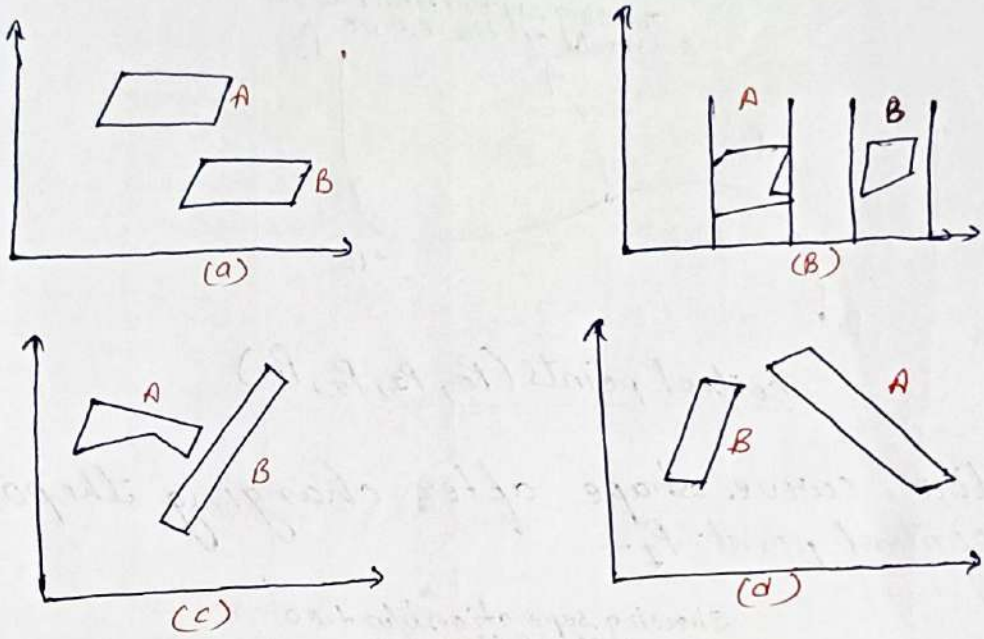
Step1: Start Algorithm

Step2: Sort all polygons by z value keep the largest value of z first.

Step3: Scan converts polygons in this order:

1. Does A is behind and non-overlapping B in the dimension of Z as shown in fig(a).
2. Does A is behind B in z and no overlapping in x or y as shown in fig(b).
3. If A is behind B in z and totally outside B with respect to view plane as shown in fig(c).
4. If A is behind B in Z and B is totally inside A with respect to view plane as shown in fig(d)

The success of any test with single overlapping polygon allows it to be painted.



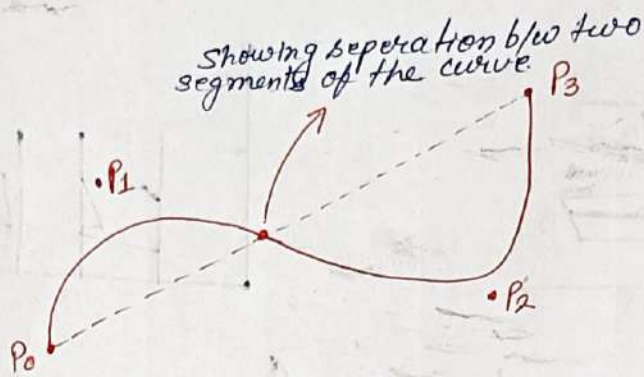
Quiz:- Explain B-Spline Curve.

Solⁿ:- Concept of B-spline curve came to resolve the disadvantages having Bezier Curve, as we all know both curves are parametric in nature. In Bezier curve we face a problem, when we change any of the control point respective location the whole curve shape gets change. But here in B-Spline curve, the only a specific segment of the changing of the corresponding location of the control points.

In B-spline curve, the control points impart local control over the curve-shape rather than the global control like Bezier-curve.

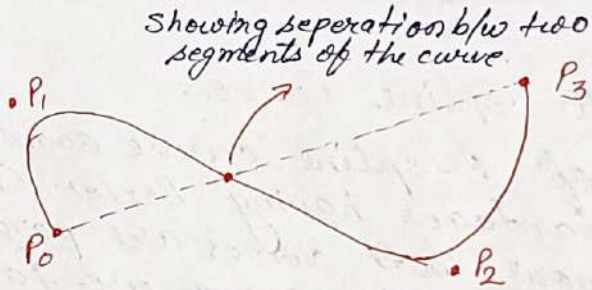
Repeat in 2013,2015,2019
B-Spline Curve

B-spline curve shape before changing the position of control P_1 –



control points (P_0, P_1, P_2, P_3)

B-spline curve shape after changing the position of control point P_1 –



Control points (P_0, P_1, P_2, P_3)

You can see in the above figure that only the segment-1st shape as we have only changed the control point P_1, and the shape of segment-2nd remains intact.

Attributes of this curve are –

- We have "n+1" control points in the above, so, $n+1 = 8$, so $n = 7$.
- Let's assume that the order of this curve is 'k' so the curve that we get will be of a polynomial degree of "k-1". Conventionally it's said that the value of 'k' must be in the range: $2 \le k \le n+1$. So lets us assume $k = 4$, so the curve degree will be $k - 1 = 3$.

- The total number of segments for this curve will be calculated through the following formula

Total no. of seg. $= n-k+2 = 7-4+2 = 5.$

Segments	Control points	Parameter.
S_0	P_0, P_1, P_2, P_3	$0 \le t \le 2$
S_1	P_1, P_2, P_3, P_4	$2 \le t \le 3$
S_2	P_2, P_3, P_4, P_5	$3 \le t \le 4$
S_3	P_3, P_4, P_5, P_6	$4 \le t \le 5$
S_4	P_4, P_5, P_6, P_7	$5 \le t \le 6$

Knots in B-spline Curve:- The point between two segments of a curve that joins each other such points are known as knots in B-spline curve. In the case of the cubic polynomial degree curve, the knots are "n+4". But in other common cases, we have "n+k+1" knots. So, for the above curve, the total knots vectors will be —

$$\rightarrow n+k+1 = 7+4+1 = 12.$$

These knot vectors could be of three types -

- Uniform (periodic)
- Open-Uniform
- Non-Uniform

B-spline Curve Equation:-

$$Q(t) = \sum_{i=0}^{n} P_i * N_{i,k}(t)$$

Where $N_{i,k}$ is Basic function of B-spline curve, P_i, k, t correspondingly represents the control points, degree, parameter of the curve.

$$N_{i,k}(t) = \frac{(t - x_i) * N_{i,k-1}(t)}{x_{i+k-1} - 1} + \frac{(x_{i+k} - t) * (N_{i+1,k-1}(t))}{x_{i+k} - x_{i+1}}$$

And following are some conditions for x_i are as follows —

$x_i = 0$; if $i < k$;

$x_i = i - k + 1$; if $k \le i \le n$;

$x_i = 0$; if $i > n$.

Some cases of Basic function:

$$N_{i,k}(t) = \begin{pmatrix} 1; \text{ if } x_i \le t \le x_{i+1} \\ 0; \text{ else} \end{pmatrix}$$

Wher $t_{min} \le t \le t_{max}$.

Properties of B-spline Curve:-

- Each basis function has 0 or +ve value for all parameters.
- Each basis function has one maximum value except for $k=1$
- The degree of B-spline curve polynomial does not depend on the number of control points which makes it more reliable to use than Bezier curve.
- B-spline curve provides the local control through control points over each segments of the curve.
- The sum of basis functions for a given parameter is one.

Repeat in (2016 2019, 2023)

Ques. What is cubic Bezier Curve? A cube Bezier curve is defined over the control points (1,1), (2,3), (4,4) and (6,1). Calculate the parmetric mid-points of this curve and show that its gradient dy/dx is 1/7

Ans. Bezier Curve :— Bezier curves are parametric curves used frequently in modelling smooth surfaces in computer graphis and many other related fields. These curves can be scaled indefinitely. This tool is also made use of in controlling motions in animations videos. The curves are used in the time domain particularly in animation, user interface design and smoothing ~~curves~~ cursor trajectory in eye gaze controlled interfaces. The most popular Bezier curves are quadratic and cubic in nature as higher degree curves are expensive to draw and evaluate. An example of the equation of Bezier curve involving two points (linear curve) is as follows:

$$B(t) = (1-t)^3 \cdot P_0 + 3(1-t)^2 \cdot t \cdot P_1 + 3(1-t) \cdot t^2 \cdot P_2 + t^3 \cdot P_3$$

where:—

$B(t)$ is the position of the curve at parameter t

P_0, P_1, P_2 and P_3 are the control points.

$(1-t)^3$, $3(1-t)^2 \cdot t$, $3(1-t) \cdot t^2$ and t^3 are the Bernstein basis polynomials.

Given the control points $P_0(1,1)$, $P_1(2,3)$, $P_2(4,4)$ and $P_3(6,1)$, we can use the parametric equation of a cubic Bezier curve.

$$B(t) = (1-t)^3 \cdot P_0 + 3(1-t^2) \cdot t \cdot P_1 + 3(1-t) \cdot t^2 \cdot P_2 + t^3 \cdot P_3$$

Now let's calculate the co-ordinates of mid point at $t = 0.5$:

$$\rightarrow B(0.5) = (1-0.5)^3 \cdot P_0 + 3(1-0.5)^2 \cdot 0.5 \cdot P_1 + 3(1-0.5) \cdot (0.5)^2 \cdot P_2 + (0.5)^3 \cdot P_3$$

$$\rightarrow B(0.5) = 0.125 \cdot (1,1) + 3 * 0.25 * (2,3) + 3 * (0.5) * 0.25 * (4,4) + 0.125 * (6,1)$$

$$\rightarrow B(0.5) = (0.125, 0.125) + (1.5, 2.25) + (0.75, 0.75) + (0.75, 0.125)$$

$$\rightarrow B(0.5) = (3.125, 3.25)$$

So, the parametric midpoint of the Bezier curve is $(3.125, 3.25)$.

To find dy/dx, we differentiate the parametric equation with respect to t and the evaluate it at $t = 0.5$. Let's ~~denote~~ denote $B(t) = (x(t), y(t))$:

$$x_t = (1-t)^3 \cdot 1 + 3(1-t)^2 \cdot t \cdot 2 + 3(1-t) \cdot t^2 \cdot 4 + t^3 \cdot 6$$

$$= y_t = (1-t)^3 \cdot 1 + 3(1-t)^2 \cdot t \cdot 3 + 3(1-t) \cdot t^2 \cdot 4 + t^3 \cdot 1$$

Differentiate both equations with respect to t, we get:

$$\frac{dx}{dt} \rightarrow -3(1-t)^2 \cdot 1 + 3(1-t)^2 \cdot 2 - 6(1-t)t + 3(1-t)t^2 + 3t^2 \cdot 4 + 3t^2 \cdot 6$$

$$\frac{dy}{dt} = -3(1-t)^2 \cdot 1 + 3(1-t)^2 \cdot 3 - 6(1-t)t + 3(1-t)^2 + 3t^2 \cdot 4 + 3t^2 \cdot 1$$

Evaluate these derivatives at $t = 0.5$ to get the slope dy/dx which is equals to $1/1$.

Q. To show that the co-ordinate transformation resulting from applying a 45-degree clock wise rotation, followed by a rotation about the x-axis, and finally counter-clockwise rotation by 45 degree is equivalent to the transformation obtained from Bresenham's algorithm.

Sol^n: 1. 45° degree clockwise rotation about the origin:-

$$R_{clockwise} = \begin{bmatrix} \cos(-\theta) & \sin(-\theta) \\ -\sin(-\theta & \cos(-\theta) \end{bmatrix}$$

For $\theta = 45^\circ$, we have:

$$R_{clockwise} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

2. Rotation about the x-axis.
This transformation doesn't affect the y-coordinate. The transformation matrix is the identity matrix:

$$R_{x\text{-axis}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

3. Counter-clockwise rotation by 45 degree about the origin:

The transformation matrix for a counter-clockwise rotation is:-

$$R_{counter\text{-}clockwise} = \begin{bmatrix} \cos 45^\circ & \sin 45^\circ \\ -\sin 45^\circ & \cos 45^\circ \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

To obtain the combined transformation matrix, we multiply these matrices in sequence.

$= R_{counter\text{-}clockwise} \cdot R_{x\text{-}axis} \cdot R_{clockwise}$

multiplying these matrices, we get:

$$T = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

After the multiplication, you will find that the resulting transformation matrix is the identity matrix which indicates that the transformation doesn't change the co-ordinates.

Ques. Explain an even odd method used for determining the interior points of polygon 2014

Solⁿ:— The even-odd rule is a method used to determine whether a point lies inside or outside a polygon. It's particularly useful in computer graphics and computational geometry for tasks such as polygon filling and point-in-polygon tests.

Here's how the even-odd rule works:

① Definition:— Consider a point (P) and a polygon (P, 1P, 2P, 3P ---- nP). to determine if the point (P) lies inside the polygon, draw a horizontal ray from P towards the right direction.

② Count Intersections:— Count the number of times the ray intersects with the edges of the polygon.

③ Rule Application:— • If the number of intersections is odd, the point (P) is inside the polygon.

• If the number of intersections is even the point P is outside the polygon.

④ Special Cases:— • If the ray passes through a vertex of the polygon, it's typically counted as a single intersection.

• If the ray passes exactly through an edge (i.e., coincides with an edge), it's usually considered to intersect only once.

(5) Example :- Let's say we have a polygon with five vertices, and we're testing whether a point lies inside it. If the ray intersects the polygon's edges three times, the point is inside. If it intersects four times, the point is outside.

(6) Implementation :- • To implement the even-odd rule algorithmically, you would iterate each edge of the polygon and check for intersections with ray using line intersection algorithms like Bresenhams algorithm or even more robust techniques like line-line intersection formulae.

• By keeping track of the number of intersections you can easily determine whether the point lies inside or outside the polygon.

The even-odd rule is simple to implement and effecient for determining the interior points of a polygon. It's widely used in computer graphics applications for tasks such as polygon filling and hit-test testing.

(2026-May) Qno.7

Que 1 (2016) for 10 x 10 frame buffer, interpret the Bresenham algorithm to find which pixels are turned on for the line segment (1,2), and (7,6)

1. First we need to calculate the differences in the X and y co-ordinates
$\Delta x = 7 - 1 = 6$
$\Delta y = 6 - 2 = 4$

2. Then, we calculate the decision parameter, which helps determine which pixel to choose at each step: Let's denote the decision parameter as p.

3. Initialize the decision Parameter: $P = 2 \times \Delta y - \Delta X \; P = 2 \times 4 - 6 \; P = 2$

4. Now, we iterate through the x co-ordinate from 1 to 7 and determine which pixels to illuminate based on the decision parameter.

- At (1,2), turn on the pixel
- At (2,3). turn on the pixel
- At (3,3); turn on the pixel
- At (4,4), turn on the pixel
- At (5 5), turn on the pixel
- At (6,5), turn on the pixel
- At (7,6), turn on the pixel

So the pixel that are turned on for the line segment from (1,2) to (7,6) are (1,2) (2,3) (3,3), (4,4) (5,5), (6,5), and (7,6).

Step 3 - • $P_{k+1} = P_k + 2\Delta y - 2\Delta x$

$= 2 + 8 - 12$

$P_{k+1} = -2$

Qno.7 Solution in detail

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

$y_{k+1} = y_{k+1} = 2+1 = 3$

• $P_{k+1} = P_k + 2\Delta y$

$= -2 + 8 = 6$

$x_{k+1} = x_{k+1} = 2+1 = 3$

$y_{k+1} = y_k = 3$

- $P_{k+1} = P_k + 2\Delta y - 2\Delta x = 6 + 8 - 12 = 2$

  $x_{k+1} = x_{k+1} = 3+1 = 4$

  $y_{k+1} = y_{k+1} = 4$

- $P_{k+1} = 2 + 8 - 12 = -12$

  $x_{k+1} = 5$

  $y_{k+1} = 5$

- $P_{k+1} = -2 + 8 = 6$, $y_{k+1} = 5$ $\quad x_{k+1} = 6$

- $P_{k+1} = 6 + 8 - 12 = 2$ $\quad y_{k+1} = 6$ $\quad x_{k+1} = 7$

② Scale the square ABCD [A(0,0), B(3,0), C(3,3), D [0,3]] $\quad$ x-Axis = 3 $\quad$ y-Axis = 3
(2017)

2017

For coordinate A(0,0)

Let the new coordinates of centre A after scaling $= (x_{new}, y_{new})$

Applying the scaling eq^n, we have

- $x_{new} = x_{old} * S_x = 0 \times 3 = 0$

  $y_{new} = y_{old} * S_y = 0 \times 3 = 0$

Thus new, coordinates after scaling (0,0)

for coordinate B(3,0)

$X_{new} = 3 \times 3 = 9$

$Y_{new} = 0 \times 3 = 0$ (9,0)

for coordinate C(3,3)

$X_{new} = 3 \times 3 = 9$

$Y_{new} = 3 \times 3 = 9$ (9,9)

for coordinate D(0,3)

$X_{new} = 0 \times 3 = 0$

$Y_{new} = 3 \times 3 = 9$ (0,9)

New coordinates

A(0,0), B(0,9), C(9,9), D(0,9)

③ (2017) 45° Rotation A(0,0) B(1,1) & C(5,2) $\theta = 45°$

for coordinate A(0,0)

2017-Qno.12

let new coordinate = (X_{new}, Y_{new})

$X_{new} = X_{old} * \cos\theta - Y_{old} * \sin\theta$

$= 0 \times \frac{1}{\sqrt{2}} - 0 \times \frac{1}{\sqrt{2}} = 0$

$Y_{new} = X_{old} * \sin\theta + Y_{old} * \cos\theta$

$= 0 \times \frac{1}{\sqrt{2}} + 0 \times \frac{1}{\sqrt{2}} = 0$

New coordinate (0,0)

for coordinate B(1,1)

$X_{new} = X_{old} * \cos\theta - Y_{old} * \sin\theta$

$= 1 \times \frac{1}{\sqrt{2}} - 1 \times \frac{1}{\sqrt{2}} = 0$

$Y_{new} = 1 \times \frac{1}{\sqrt{2}} + 1 \times \frac{1}{\sqrt{2}} = \frac{2}{\sqrt{2}}$

New coordinate $\left(0, \frac{2}{\sqrt{2}}\right)$

for coordinate C(5,2)

$X_{new} = 5 \times \frac{1}{\sqrt{2}} - 2 \times \frac{1}{\sqrt{2}} = \frac{3}{\sqrt{2}}$

$Y_{new} = 5 \times \frac{1}{\sqrt{2}} + 2 \times \frac{1}{\sqrt{2}} = \frac{7}{\sqrt{2}}$

New coordinate $A[0,0]$, $B\left[0, \frac{2}{\sqrt{2}}\right]$, $C\left[\frac{3}{\sqrt{2}}, \frac{7}{\sqrt{2}}\right]$

(9) triangle A(0,0), B(4,0), C(0,4) and
(2018) Shearing transformation a=2, b=3

Shearing on X-Axis **(2018-Qno.12)**

- for coordinate A(0,0)

$X_{new} = X_{old} + Sh_x * Y_{old} = 0 + 2 \times 0 = 0$

$Y_{new} = Y_{old} = 0$

new coordinate (0,0)

- For coordinate B(4,0)

$x_{new} = 4 + 2 \times 0 = 4$

$y_{new} = 0$ coordinate (4,0)

- For coordinate C(0,4)

$x_{new} = 0 + 2 \times 4 = 8$

$y_{new} = 4$ coordinate (8,4)

Shearing in Y-Axis = 3

- $x_{new} = x_{old} = 0$

$y_{new} = y_{old} + sh_y * x_{old} = 0 + 3 \times 0 = 0$

coordinate (0,0)

- For coordinate B(4,0)

$x_{new} = 4$

$y_{new} = 0 + 3 \times 4 = 12$ = (4,12)

- For coordinate C(0,4)

$x_{new} = 0$

$y_{new} = 4 + 3 \times 0 = 4$

coordinate (0,4)

⑤ Rotate a triangle (4,6), (2,4) (6,2) vertex (2018) (4,6) by 180 clockwise.

clockwise = - θ

2018-Qno.10

for coordinate A(4,6)

let the new coordinate = (Xnew, Ynew) cos 180 = -1

$X_{new} = X_{old} \times \cos\theta - Y_{old} \times \sin\theta$

$= 4 \times 1 - 4 \times 0 \quad = 4$

$Y_{new} = X_{old} \times \sin\theta + Y_{old} \times \cos\theta$

$= 4 \times 0 + 6 \times 1 = 6 \quad (4,6)$

for coordinate B(2,4)

$X_{new} = 2 \times 1 - 4 \times 0 = 2$

$Y_{new} = 2 \times 0 + 4 \times 1 = 4 \quad (2,4)$

for coordinate C(6,2)

$X_{new} = 6 \times 1 - 2 \times 0 = 6$

$Y_{new} = 6 \times 0 + 2 \times 1 = 2 \quad (6,2)$

new coordinate A(4,6) B(2,4) C(6,2)

(6) A(3,0) B(6,2) by Brusenham's line algo. (6)

2022

Given (3,0) and (6,2)

$x_1\ y_1$ $x_2\ y_2$

Step 1 := $\Delta x = x_2 - x_1$

$= 6 - 3$

$= 3$

$\Delta y = y_2 - y_1$

$= 2 - 0$

$= 2$

Step 2 - $P_k = 2\Delta y - \Delta x$

$= 2 \times 2 - 3$

$= 4 - 3 = 1$

P_k	P_{k+1}	x_{k+1}	y_{k+1}
1	-1	4	1
-1	3	5	1
3	0	6	2
0	-2	7	3
-2	2	8	3
2	0	9	4
0	-2	10	5

Step 3 - $P_{k+1} = P_k + 2\Delta y - 2\Delta x$

$= 1 + 4 - 6 = -1$

$x_{k+1} = 3 + 1 = 4$

$y_{k+1} = 0 + 1 = 1$

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

$= -1 + 4$

$= 3$

$x_{k+1} = 4 + 1 = 5$

$y_{k+1} = 1$

(7)

- $P_{k+1} = P_k + 2\Delta y - 2\Delta x = 2 + 4 - 6 = 0$

$x_{k+1} = 5+1 = 6$

$y_{k+1} = 1+1 = 2$

- $P_{k+1} = P_k + 2\Delta y - 2\Delta x = 0 + 4 - 6 = -2$

$x_{k+1} = 6+1 = 7$

$y_{k+1} = 2+1 = 3$

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

$= -2 + 4 = 2$

$x_{k+1} = 7+1 = 8$, $y_{k+1} = 3$

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

$= 2 + 4 - 6 = 0$

$x_{k+1} = 8+1 = 9$, $y_{k+1} = 3+1 = 4$

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

$= 0 + 4 - 6 = -2$

$x_{k+1} = 9+1 = 10$, $y_{k+1} = 4+1 = 5$

7 centre (4,5) radius = 4 Bresenham's algo

Procedure: (2023)

[Question No. 7, 2023-June]

Given :- centre point of the circle (4,5)

Radius of circle R = 4

Step 1 - $x_0 = 0$

$y_0 = R = 4$

Step 2 $P_k = 3-2R$

$P_k = 3-2\times4$

$= 3-8$

$P_k = -5$

P_k	P_{k+1}	(x_{k+1}, y_{k+1})
-5	21	(5,5)
21	31	(6,6)
31	41	(7,7)
41	51	(8,8)

Step 3 - • $P_k = -5$

$x_k = x_k + 1 = 4+1 = 5$

$y_k = y_k = 5$

$P_{k+1} = P_k + 4x_{k+1} + 6$

$= -5 + 4\times5 + 6$

$= -5 + 20 + 6$

$= 21$

• $P_k = 21$

$x_k = x_k + 1 = 5+1 = 6$

$y_k = y_k + 1 = 5+1 = 6$

$P_{k+1} = P_k + 4(x_{k+1} - y_{k+1}) + 10$

$= 21 + 4(5-5) + 10 = 21 + 10 = 31$

- $P_k = 31$

$x_{k+1} = 6+1 = 7$

$y_{k+1} = 6+1 = 7$

$P_{k+1} = P_k + 4(x_{k+1} - y_{k+1}) + 10$

$= 31 + 4(7-7) + 10 = 41$

- $P_k = 41$

$x_{k+1} = 7+1 = 8$

$y_{k+1} = 7+1 = 8$

$P_{k+1} = P_k + 4(x_{k+1} - y_{k+1}) + 10$

$= 41 + 4(7-7) + 10$

$= 41 + 10 = 51$

Perform a 45° Rotation of triangle ABCD when A(0,0) B(1,1), C(5,2) about origin [2017 Q-No. 12 Solve by Matrix]

Sol. Transformation matrix = Transmition Rotation (θ) Invers transl

$$\begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & dy \\ 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -dx \\ 0 & 1 & -dy \\ 0 & 0 & 1 \end{bmatrix}$$

$dx = 0 \quad \theta = 45°$

$dy = 0$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \cos 45 & -\sin 45 & 0 \\ \sin 45 & \cos 45 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 5 \\ 0 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & \frac{\sqrt{2}}{2} + (-\frac{\sqrt{2}}{2}) & 5\frac{\sqrt{2}}{2} + (-\frac{2\sqrt{2}}{2}) \\ 0 & \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} & 5\frac{\sqrt{2}}{2} + 2\frac{\sqrt{2}}{2} \\ 1 & \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} + 1 & 5\frac{\sqrt{2}}{2} + 2\frac{\sqrt{2}}{2} + 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & \frac{3\sqrt{2}}{2} \\ 0 & \frac{\sqrt{4}}{2} & 7\frac{\sqrt{2}}{2} \\ 0 & 1+\frac{\sqrt{4}}{2} & 1+7\frac{\sqrt{2}}{2} \end{bmatrix}$$

$A'(0,0,)$

$B'(0, \frac{\sqrt{4}}{2})$

$C'(\frac{3\sqrt{2}}{2}, 7\frac{\sqrt{2}}{2})$

write a procedure from Bresenhams line drawing algorithm for draw a line using Bresenhams line algorithm for the end points (20,10) to (30,18).

2015-Qno.9

Given- (20,10) or (30,18)
x_1, y_1 x_2, y_2

Step1) $\Delta x = x_2 - x_1 = 30 - 20 \Rightarrow \boxed{10 = \Delta x}$

$\Delta y = y_2 - y_1 = 18 - 10 \Rightarrow \boxed{8 = \Delta y}$

Step2) $P_K = 2\Delta y - \Delta x$

$= 2 \times 8 - 10$

$= 16 - 10$

$= \boxed{6}$

graph

Step3) $P_K >= 0$

$P_{K+1} = P_K + 2\Delta y - 2\Delta x$

$X_{K+1} = X_K + 1$

$Y_{K+1} = Y_K + 1$

① $P_K = 6$

$P_{K+1} = 6 + 2 \times 8 - 2 \times 10$

$= 6 + 16 - 20$

$= 22 - 20 \Rightarrow \boxed{2}$

$X_{K+1} = 20 + 1 = 21$

$Y_{K+1} = 10 + 1 = 11$

② $P_K = 2$

$P_{K+1} = 2 + 2 \times 8 - 2 \times 10$

$= 2 + 16 - 20$

$= 18 - 20$

$= \boxed{-2}$

P_K	P_{K+1}	X_{K+1}	Y_{K+1}
6	2	21	11
2	-2	22	12
-2	14	23	12
14	10	24	13
10	6	25	14
6	2	26	15
2	-2	27	16
-2	14	28	16
14	10	29	17
10	6	30	18

$x_{k+1} = 21+1 = 22$

$y_{k+1} = 11+1 = 12$

(3) $P_k = -2$

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

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

$y_{k+1} = y_k$

$P_{k+1} = -2 + 2\times 8$

$= -2 + 16 = \boxed{14}$

$x_{k+1} = 22+1 = 23$

$y_{k+1} = 12$

(4) $P_k = 14$

$P_{k+1} = 14 + 2\times 8 - 2\times 10$

$= 14 + 16 - 20$

$= 30 - 20 \Rightarrow \boxed{10}$

$x_{k+1} = 23+1 = 24$

$y_{k+1} = 12+1 = 13$

(5) $P_k = 10$

$P_{k+1} = 10 + 2\times 8 - 2\times 10$

$= 10 + 16 - 20$

$= 26 - 20 \Rightarrow \boxed{6}$

$x_{k+1} = 24+1 = 25$

$y_{k+1} = 13+1 = 14$

(6) $P_k = 6$

$P_{k+1} = 6 + 2\times 8 - 2\times 10$

$= 6 + 16 - 20$

$= 22 - 20 \Rightarrow \boxed{2}$

$x_{k+1} = 25+1 = 26$

$y_{k+1} = 14+1 = 15$

(7) $P_k = 2$

$P_{k+1} = 2 + 2\times 8 - 2\times 10$

$= 2 + 16 - 20$

$= 18 - 20 \Rightarrow \boxed{-2}$

$x_{k+1} = 26+1 = 27$

$y_{k+1} = 15+1 = 16$

(8) $P_k = -2$

$P_{k+1} = -2 + 2\times 8$

$= -2 + 16$

$= \boxed{14}$

$x_{k+1} = 27+1 = 28$

$y_{k+1} = 16$

(9) $P_k = 14$

$P_{k+1} = 14 + 2\times 8 - 2\times 10$

$= 14 + 16 - 20$

$= 30 - 20 \Rightarrow \boxed{10}$

$x_{k+1} = 28+1 = 29$

$y_{k+1} = 16+1 = 17$

(10) $P_k = 10$

$P_{k+1} = 10 + 2\times 8 - 2\times 10$

$= 10 + 16 - 20$

$= 26 - 20 \Rightarrow \boxed{6}$

$x_{k+1} = 29+1 = 30$

$y_{k+1} = 17+1 = 18$

Clip the line with end points (0,0) & (12,12) using Cohen Southerland algorithm. The clipping rectangle is defined with bottom left edge as (1,1) and top right edge as (10,10).

2015-Qno.10

Given – A(0,0) & B(12,12),
$P_1 = (1,1)$ & $P_2 = (10,10)$

Find reason code A → 0101, B → 1010

Apply AND operation

$$\begin{array}{rl} A = & 0101 \\ B = & 1010 \\ \hline AND = & 0000 \end{array}$$

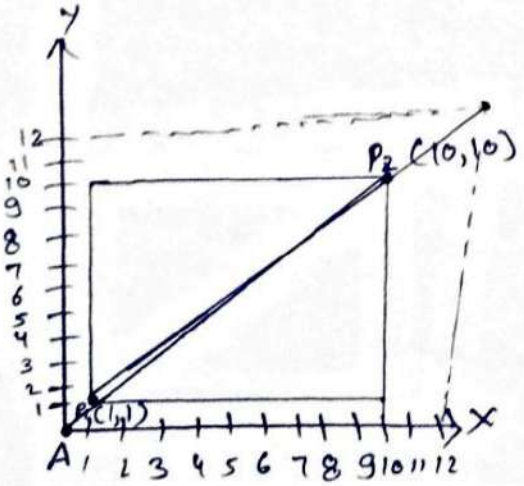
Since it is 0000 line is particially visible so Clipping is needed.

③ $m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{12-0}{12-0} = \frac{12}{12} = 1$

$y = m(x_{min} - x_1) + y_1$

$= 1(1-0) + 0$

$= 1 \times 1 + 0 = \boxed{1}$



Digitize a line from (10,12) to (15,15) on a raster screen using Brushams straight line algorithm. What are the various line drawing algorithms?

2023-Qno.6

Given - $(10,12)$ or $(15,15)$

x_1, y_1 $\quad$ x_2, y_2

Step 1)

$\Delta x = x_2 - x_1 = 15 - 10 = \boxed{5 = \Delta x}$

$\Delta y = y_2 - y_1 = 15 - 12 = \boxed{3 = \Delta y}$

Step 2)

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

$P_K = 2 \times 3 - 5$

$P_K = 6 - 5$

$P_K = \boxed{1}$

Step 3) ① $P_{K+1} = P_K + 2\Delta y - 2\Delta x$

$= 1 + 2 \times 3 - 2 \times 5$

$= 1 + 6 - 10$

$= \boxed{-3}$

$X_{K+1} = X_K + 1$

$X_{K+1} = 10 + 1 = 11$

$Y_{K+1} = Y_K + 1$

$Y_{K+1} = 12 + 1 = 13$

② $P_K = -3$

$P_{K+1} = P_K + 2\Delta y$

$= -3 + 2 \times 3$

$= -3 + 6$

$= \boxed{3}$

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

$x_{k+1} = 11 + 1 = 12$

$y_{k+1} = y_k$

$y_{k+1} = 13$

③ $P_k = 3$

$P_{k+1} = 3 + 2 \times 3 - 2 \times 5$

$= 3 + 6 - 10$

$= -1$

$x_{k+1} = 12 + 1 = 13$

$y_{k+1} = 13 + 1 = 14$

④ $P_k = -1$

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

$= -1 + 2 \times 3$

$= -1 + 6$

$= 5$

$x_{k+1} = 13 + 1 = 14$

$y_{k+1} = 14$

⑤ $P_k = 5$

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

$= 5 + 2 \times 3 - 2 \times 5$

$= 5 + 6 - 10$

$= 1$

$x_{k+1} = 14 + 1 = 15$ / $y_{k+1} = 14 + 1 = 15$

P_k	P_{k+1}	x_{k+1}	y_{k+1}
1	-3	11	13
-3	3	12	13
3	-1	13	14
-1	5	14	14
5	1	15	15

2023-Qno.11

(2023)

Que-3 find a transformation of triangle $A(1,0)$, $B(0,1)$ $C(1,1)$ by -

(a) Rotating $45°$ about the origin and then translating one unit in x and y directions.

(b) Translation one unit in x and y direction and the rotating $45°$ about the origin

Sol-(a) $R = \begin{bmatrix} \cos 45° & \sin 45° & 0 \\ -\sin 45° & \cos 45° & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$ tx ty

$$R \cdot T = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 1 & 1 & 1 \end{bmatrix} \Rightarrow \begin{matrix} A' \\ B' \\ C' \end{matrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 1 & 1 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{1}{\sqrt{2}}+1 & \frac{1}{\sqrt{2}}+1 & 1 \\ 0-\frac{1}{\sqrt{2}}+1 & 0+\frac{1}{\sqrt{2}}+1 & 0+0+1 \\ \frac{1}{\sqrt{2}}+\left(-\frac{1}{\sqrt{2}}\right)+1 & \frac{1}{\sqrt{2}}+\frac{1}{\sqrt{2}}+1 & 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \frac{1}{\sqrt{2}}+1 & \frac{1}{\sqrt{2}}+1 & 1 \\ -\frac{1}{\sqrt{2}}+1 & \frac{1}{\sqrt{2}}+1 & 1 \\ 1 & \sqrt{2}+1 & 1 \end{bmatrix}$$

$A' = \left(\frac{1}{\sqrt{2}}+1, \frac{1}{\sqrt{2}}+1\right)$

$B' = \left(\frac{-1}{\sqrt{2}}+1, \frac{1}{\sqrt{2}}+1\right)$

$C' = \left(1, \sqrt{2}+1\right)$ A

Sol (b) – $T \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$ $R = \begin{bmatrix} \sin 45^\circ & \cos 45^\circ & 0 \\ -\sin 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$$TR = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & \sqrt{2} & 1 \end{bmatrix}$$

$$\left.\begin{matrix} A' \\ B' \\ C' \end{matrix}\right\} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad P' = P \cdot T \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & \sqrt{2} & 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} + \sqrt{2} & 1 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} + \sqrt{2} & 1 \\ \frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} + \sqrt{2} & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{1}{\sqrt{2}} & 3\sqrt{2} & 1 \\ -\frac{1}{\sqrt{2}} & 3\sqrt{2} & 1 \\ 0 & 2\sqrt{2} & 1 \end{bmatrix}$$

$A' = \left(\frac{1}{\sqrt{2}}, 3\sqrt{2}\right)$

$B' = \left(-\frac{1}{\sqrt{2}}, 3\sqrt{2}\right)$

$C' = (0, 2\sqrt{2})$