IIT2019174
Jyotsana Srivastava

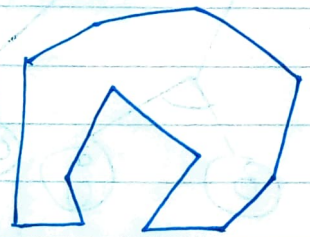
Jyotsana

Question 1

Convex polygon -

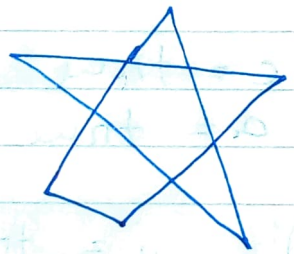
(All angles less than 180°)

Concave -

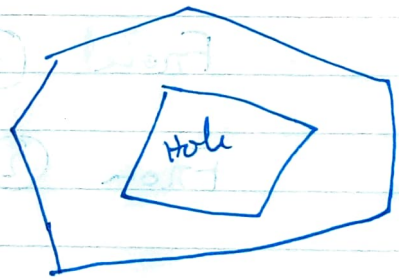


(Some interior angles more than 180°)

Intersecting -



Polygon with a hole -



Question 2

A polygon is convex if for all edges, all other vertices lie on the same side of the edge.
Otherwise it is concave.

IIT2019174
Jyotsana Srivastava

Jyotsana

Question 3

All internal angles of a convex Polygon are ~~less~~ smaller than 180°

Question 4

~~Polygons are used because polygon surfaces are common in design and~~

Polygons are used because ~~they are~~ it is very ~~easy~~ easy to describe every object as polygons and so in rendering needs to be ~~g~~ done in real-time, it is much easily and fastly done using polygons ~~as~~

Question 5

Scan line algorithm says that to define the "interior" of a polygon, we consider ~~two~~ one vertex on the left and right boundary each, and ~~color~~ fill all the pixels in between these vertices with the color of the polygon.

A major ~~cut~~ challenge in this is that in case of concave polygons, with some ~~reflex~~ edges as: 

IIT2019174

Jyotsana Srivastava

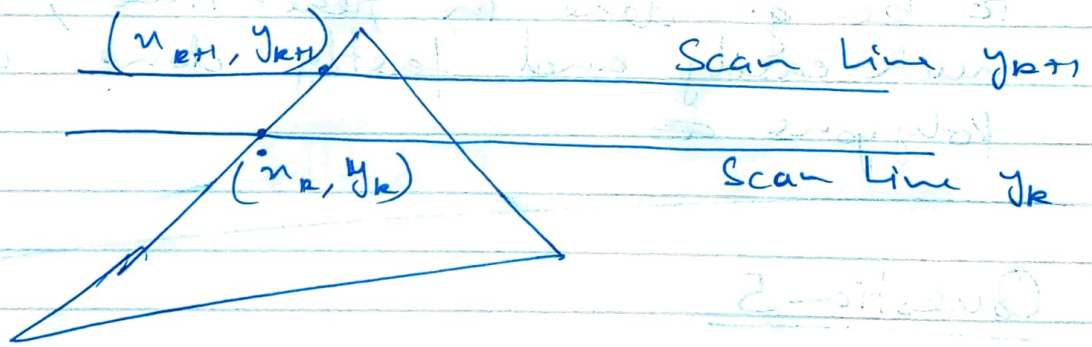
Jyotsana

this may happen:

Here, the region b/w B and C should also be considered inside the polygon, but the color reverses.

To tackle this, the point B is split into two points, and is considered as two consecutive vertices.

Question 6



$\therefore$ m = Slope

Changes in y-coordinates are considered as:

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

$$\therefore m = \frac{y_{k+1} - y_k}{x_{k+1} - x_k} = \frac{1}{x_{k+1} - x_k}$$

$$\Rightarrow \boxed{x_{k+1} = x_k + \frac{1}{m}}$$

IIT2019174
Jyotsana Srivastava

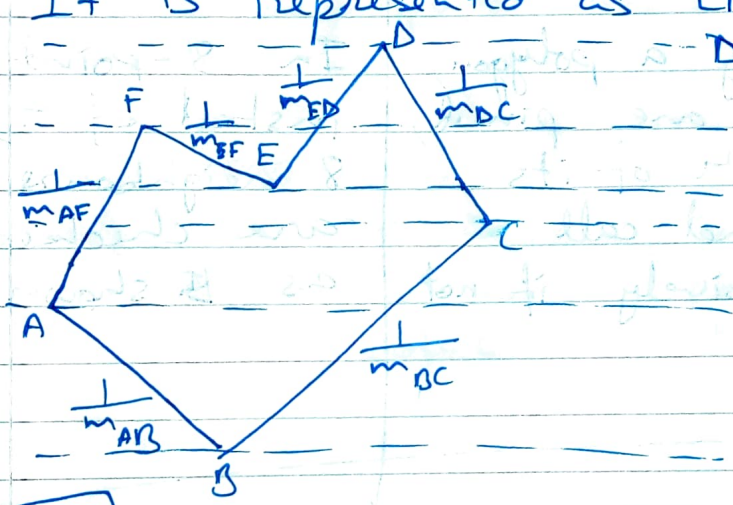
Jyotsana

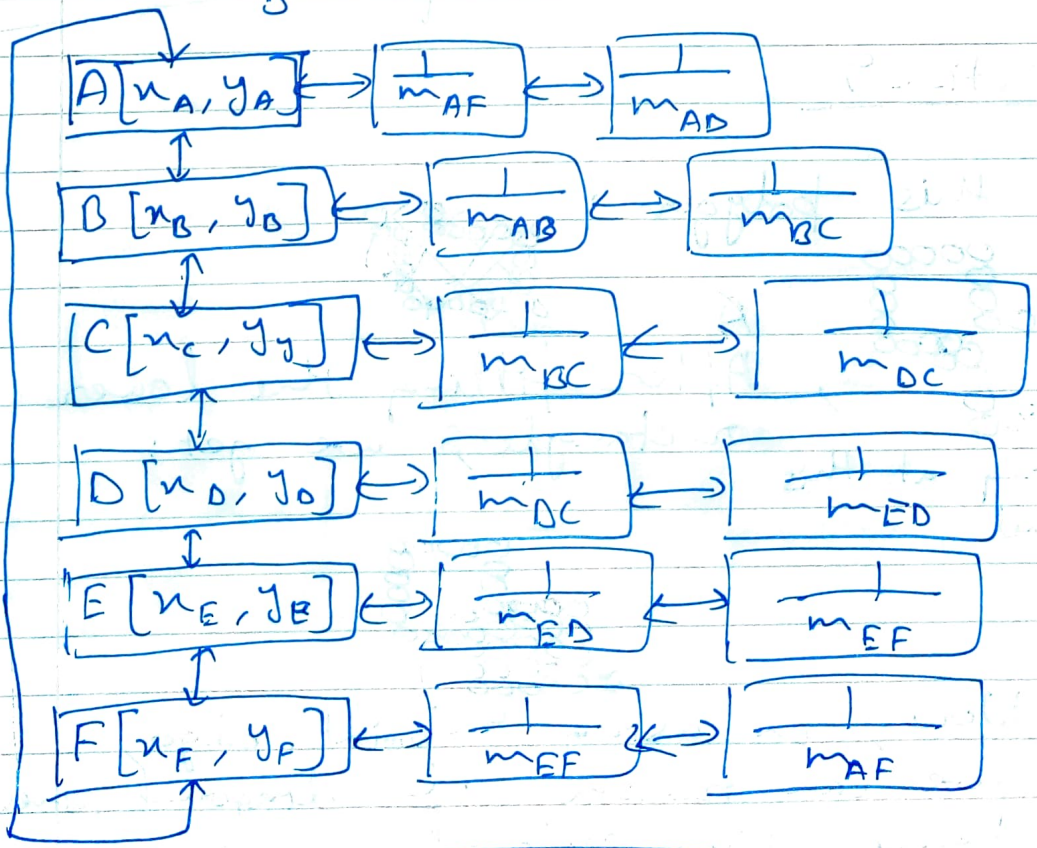
∴ After k^{th} intersection points,

$$x_k = x_0 + \frac{k}{m}$$

Question 7

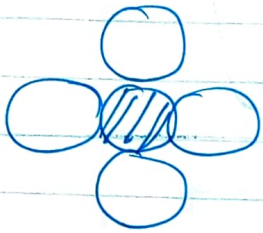
It is represented as Linked List:

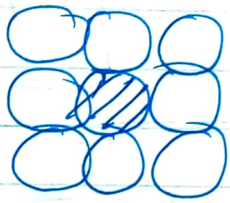




IIT2019174
Tyotsana Srivastava Tyotsana

Question 8





4-point algorithm

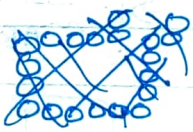
While filling a polygon, after filling one pixel, we check 4 of its neighbors and call them recursively if not filled

8-point algo.

In 8-point, instead of 4, 8 neighbours are checked as shown.

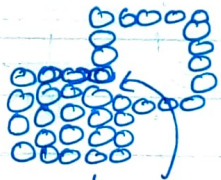
Question 9

In this polygon:



, After filling the lower rectangle, we get:

Starting at this pixel



Now, on using 4-point algo, we can reach this dragonal pixel, hence, upper rectangle remains empty.

IIT2019174
Jyotsana Srivastava

Jyotsana

Question 10

In Boundary fill algorithm, the coloring starts at a pixel inside a boundary defined area, and is filled progressively untill a boundary is reached.

Flood fill algo fills the entire connected regia of a particular color to ~~be~~ with another.

Boundary fill	Flood fill
Only process image containing single boundary color	Can process image containing more than one boundary color
Faster	Slower

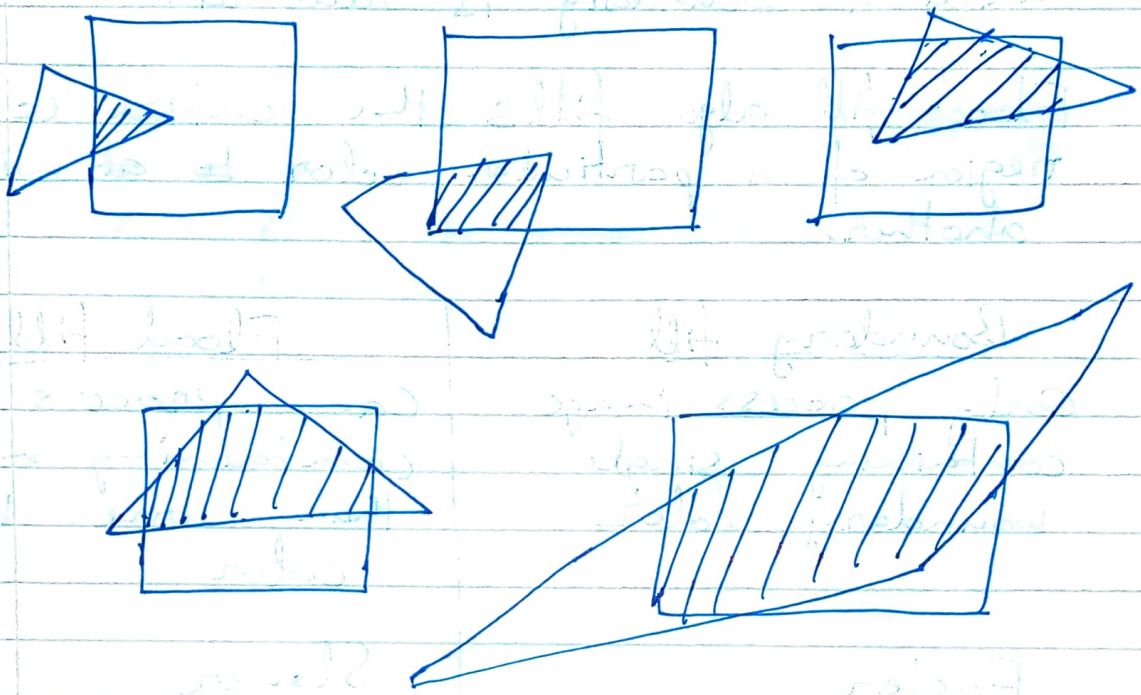
Question 11

0	1	1	1	1	1	1	0
0	0	0	0	0	1	1	0
0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0
0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

UT2019174
Jyotsana Srivastava Jyotsana

Question 12

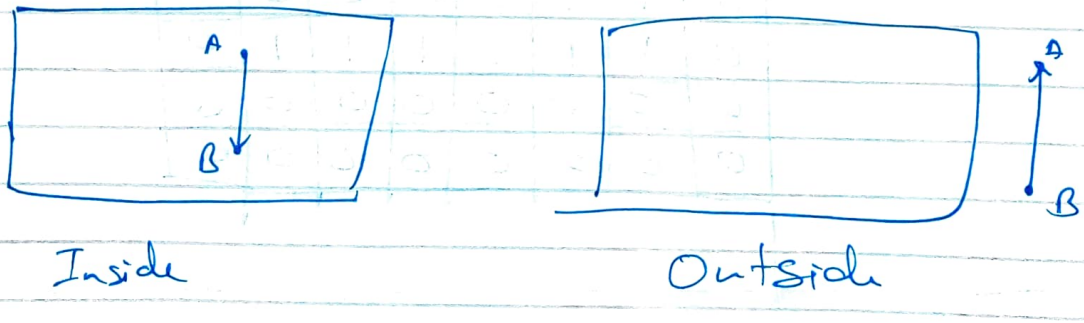
On clipping a triangle, there can become 3 to 7 edges in the new polygon.



Question 13

There will be no o/p if both vertices of the edge are either inside or outside the polygon.

Eg:



IIT2019174
Jyotsana Srivastava Jyotsana

Question 14

Let the point be p

Let the plane be defined by a point on the plane: q
and a normal to the plane: n

Then,

$(p-q) \cdot n < 0$ if p is inside the plane

$(p-q) \cdot n = 0$ p - on the plane

$(p-q) \cdot n > 0$ p - outside the plane

Question 15

Its basic routine is as follows:

1. Go around the polygon one ~~vert~~ vertex at a time
2. Current vertex ~~p~~ has position p
3. Previous vertex has position s, and it has been added to the output if appropriate.

IIT2019174
Jyotsana Srivastava

Now, check for 4 cases:

- s - inside, p - inside :
  Add p to o/p
- s - inside, p - outside :
  Find intersection point i, and add it to o/p
- s - outside, p - outside :
  Add nothing
- s - outside, p - inside :
  Find intersection point i.
  Add i to o/p, followed by p

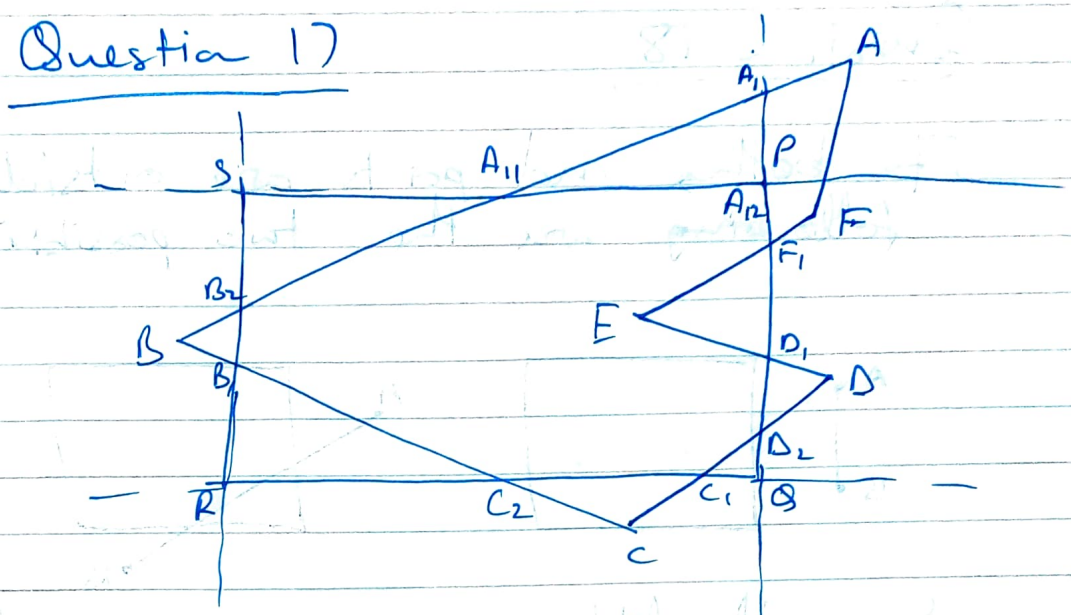
Question 16

Because, finally, only the portion withing the display window will be visible, so, it will be a wastage of resources if we don't clip and fill the invisible parts of the polygon as well.

11T2019174
Jyotsna Srivastava

Jyotsna

Question 17



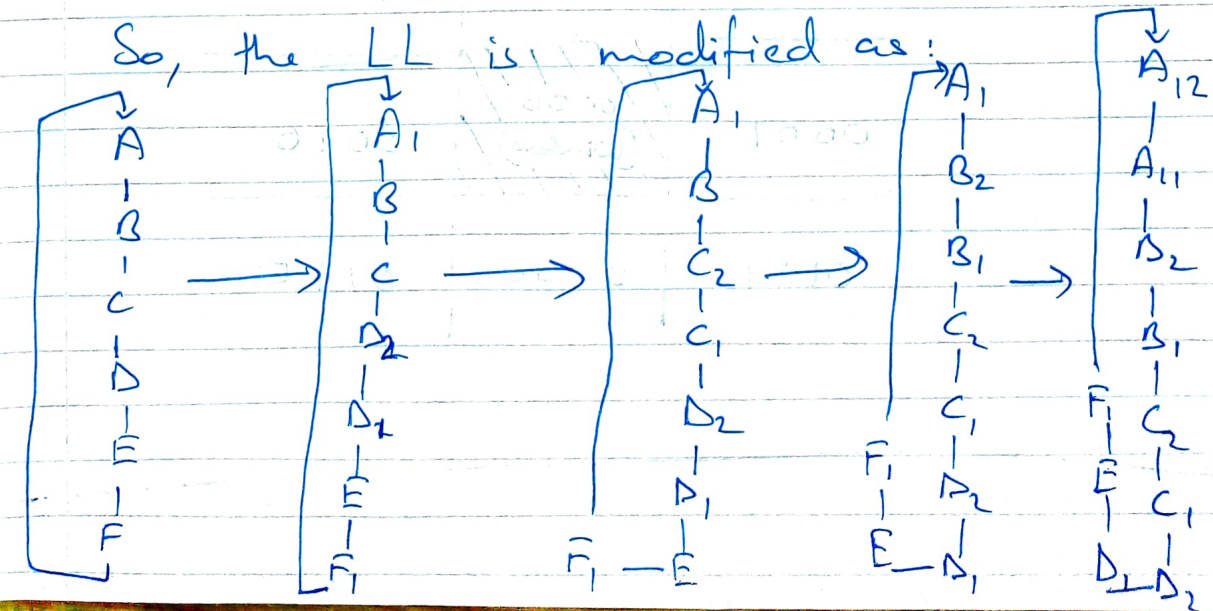
On clipping against S window edge PQ, we get intersection points A_1, F_1, D_1, D_2

Then clipping against edge QR, we get C_1, C_2

Clipping against RS, we get B_1, B_2

Finally, on clipping against SR, we get A_{11}, A_{12}

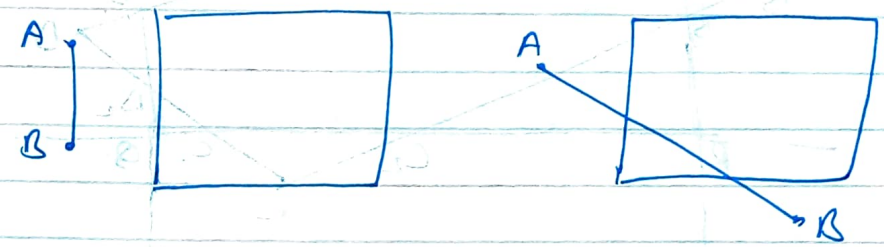
So, the LL is modified as:



IIT2019174
Jyotsana Srivastava

Question 18

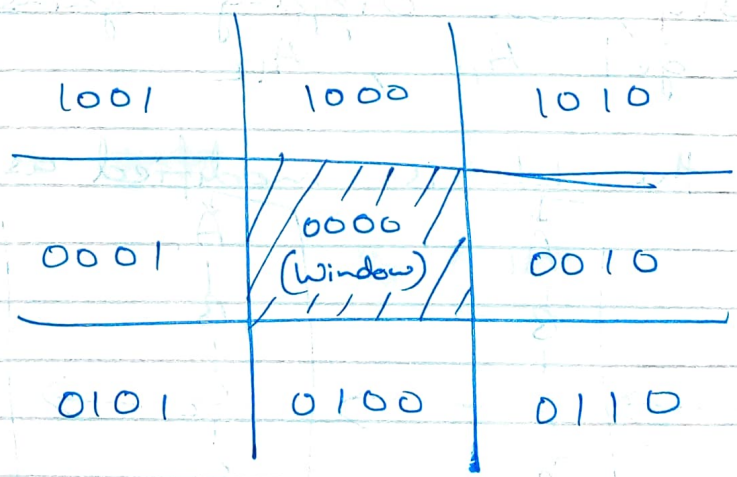
If both end points are outside, following are the two possibilities:



So, clipping may or may not be required.

Question 19

Positions of vertices are remembered as a 4-bit code denoting their position in one of the 9 regions as defined:



11726949174

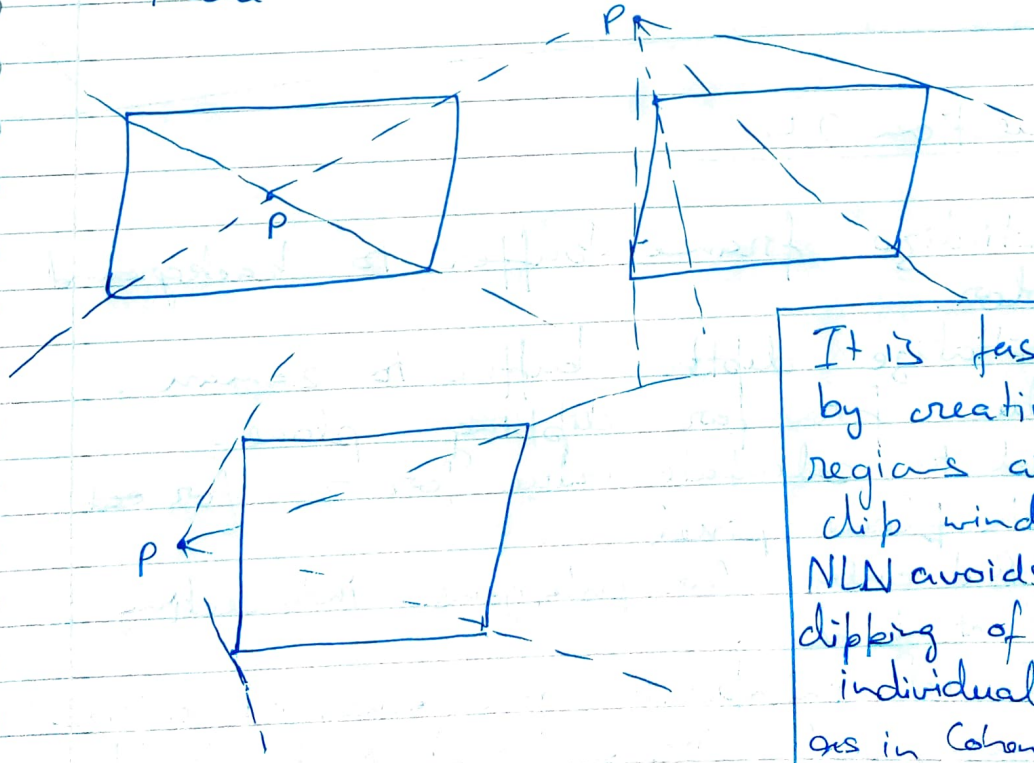
Systems Simulation

Question 20

- If the 4-bit value of both the vertices of the edge are 0000, then they are trivially accepted.
- Else if bitwise AND (i.e., v1 & v2 != 0) is non-zero, then they are trivially rejected.
- ~~Otherwise - they need~~

Question 21

As opposed to Cohen-Sutherland Clipping, in NLN, the region is divided based on a point (P).
There are 3 alternatives to do so:



It is faster as, by creating more regions around the clip window, the NLN avoids multiple clipping of an individual line segment as in Cohen-Sutherland.

IIT2019174
Jyotsana Srivastava
Jyotsana

Question 22

If N is the normal to a ~~sur~~ surface, and V is the viewing vector.

Then,

$N \cdot V > 0 \Rightarrow$ Back surface

$N \cdot V < 0 \Rightarrow$ Front surface

Question 23

1. Primitive lies outside field of view
2. Primitive is back-facing
2. Primitive is ~~occupied~~ occluded by one or more objects nearer the viewer.

Question 24

1. Initialize frame buffer to background color.
2. Initialize depth buffer to z=max. value for far clipping plane.
3. Need to calculate value for z for ~~eat~~ each ~~plane~~ pixel
   - But only for polygons intersecting that pixel.
   - Could interpolate from values at vertices
4. Update both frame and depth buffer.

IIT2019174
Jyotsana Srivastava

Question 25

Z-buffer and Ray casting are not mutually exclusive. Z-buffers are frequently used to support triangle-based, back-projected rendering, and they can also be used to support ray tracing.

Question 26

1. One of class of "list priority" algorithms returns ordered list of polygon fragments for specified view point (static pre-processing stage).
2. Choose polygon arbitrarily.
3. Divide scene into front (relative to normal) and back half-spaces.
4. Split any polygon lying on both sides
5. Choose a polygon from each side - split scene again.
6. Recursively divide each side until each node contains only 1 polygon.

IIT2019117 Jyotsana Srivastava

Question 27

Start with root window:

1. If zero or one intersecting, contained or surrounding polygon then scan convert window.
2. Else subdivide window as quadtree
3. Recurse until zero or one polygon, or some set depth.
4. Depth may be pixel resolution, display nearest polygon.

Question 28

Lets assume clip window is rectangle in shape.

Consider Minimum value : (X_{Wmin}, Y_{Wmin})

and Maximum value: (X_{Wmax}, Y_{Wmax})

Points will be clipped on the following basis –

$$X_{Wmin} \leq x \leq X_{Wmax}$$
$$\&\&$$
$$Y_{Wmin} \leq y \leq Y_{Wmax}$$

} Don't clip, since inside.

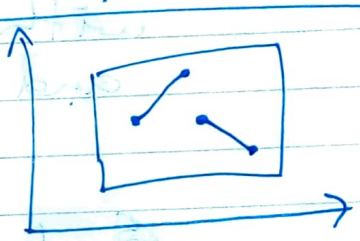
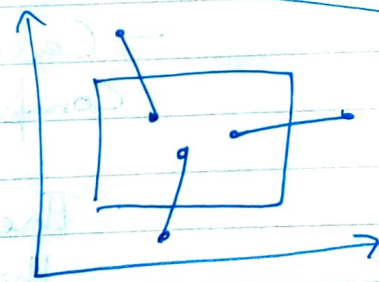
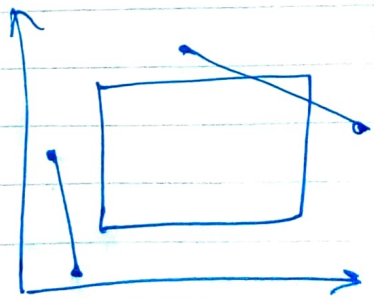
1/T2014/74 Jyotsna Srivastava

In 3D the cuboid is

$$X_{Wmin} \leq x \leq X_{Wmax}$$
$$Y_{Wmin} \leq y \leq Y_{Wmax}$$
$$Z_{Wmin} \leq z \leq Z_{Wmax}$$

} Don't clip since inside the ~~cub~~ cuboid.

Question 29

Situation	Solution	Example
Both end-points inside the window	Don't clip	
One end-point inside and the other ~~outise~~ outside the window	Must clip	
Both end-points outside the window	Don't know!	

IIT2019172 Jyotsana Sriramulam

Question 30

Brute Force can be performed as:

- Don't clip if both ends are within the window
- When one end-point outside and one inside: calculate the intersection point using line equation and clip from this point.
- For line with both end-points outside, test the line for intersection with all of the window boundaries and clip appropriately.

Drawback:

- Calculating line intersections is computationally expensive.
- There can be many lines, so this method is too slow.

IIT 2019 (7th) Jyotsana Srivastava

Question 31

C is (4, 1)

A → (4, 1+5) = (4, 6)

B → (4+8, 1+5) = (12, 6)

Equation of line AB:

$$y - 6 = 0 \cdot (x - y) \Rightarrow \boxed{y = 6} \quad \text{---- (1)}$$

Equation of line XY:

$$y - 10 = \frac{-8 \times 2}{13}(x - 5)$$

$$\Rightarrow \boxed{y - 10 = \frac{-16}{13}(x - 5)} \quad \text{---- (II)}$$

Putting equⁿ (1) in equⁿ (2):

$$-4 = \frac{-16}{13}(x - 5) \Rightarrow x = \frac{13}{4} + 5$$

$$\Rightarrow \boxed{x = 8.25}$$

$$\therefore \boxed{(x', y') = (8.25, 6)}$$

IIT2019174
Jyotsana Srivastava

Question 32

For each (edge, window-edge) pair, we need to perform 3+2+1 = 6 operations

∴ Total = 3 × 4 × 6
= 72 mathematical operations for 3 edges.

Question 33

Advantages:

- Simple to implement in hardware.
- Diversity of primitives
- Unlimited scene complexity
- Don't need to calculate object - object intersections.

Disadvantages:

- Extra memory and bandwidth
- Wast time drawing hidden objects
- Z-precision errors
- May have to use point sampling.

IIT2019174 Jyotsana Srivastava

Question 34

- Back Face removal algorithm
- 2-buffer algo
- Painter's algo
- Scan line algo
- Subdivision algo
- Floating horizon algo

Question 35

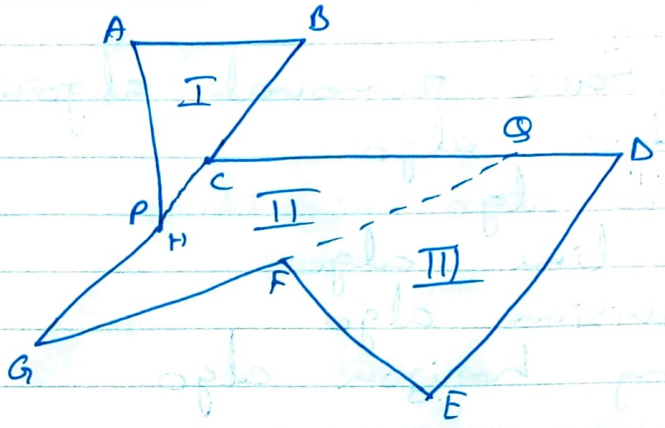
There are two approaches for removing hidden surface problem.

① Object space method:
implemented in physical coordinate system

② Image space method:
Implemented in screen coordinate system.

IIT2019174 Jyotsana Sreedharan

Question 36



Question 37

Question 38

Start with the root window.

1. If zero or one intersecting, contained or surrounding polygon then scan convert window.
2. Else subdivide window as quadtree
3. Recurse until zero or one polygon, or some set depth
4. Depth may be pixel resolution, display nearest polygon.

IIT2019174
Jyotsana Srivastava

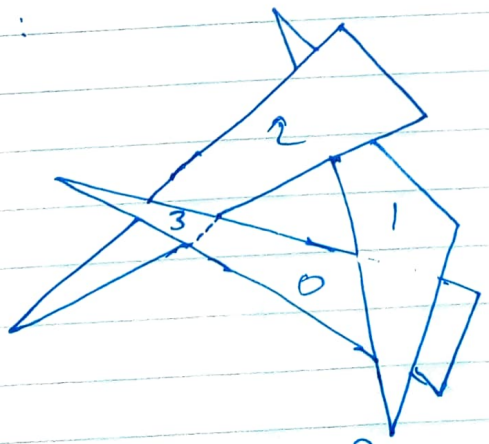
Question 39

The painter's algorithm is called as a ~~priority~~ priority fill, is one of the easiest results to ~~be~~ the visibility issue in 3D graphics.
When projecting a 3D view onto a 2D screen, it is essential at various points to be finalized which polygons are visible, and which are hidden

Question 40

In this situation, any one of the triangles are ~~to~~ broken into two, ~~and~~ with one part considered below and other above. ~~for~~.

Here, we can break the blue triangle as:



So, the part 3 can be considered as top.