→ Depth- Buffer (Z-buffer method) :-

↳ image-space approach

The basic idea is to test the Z-depth of each surface to determine the closest (visible) surface.

Frame buffer

↓

used to store the pixel intensity values of color at each position (x,y)

↓

The pixel intensity / color will be given for each surface.

Depth Buffer

(used to store value of z)

↳ depth of (x,y) point

(IMP)

Now, the size of the frame buffer as well as the depth buffer will be given in the problem as m x n.

* Advantages :-

→ It processes one object at a time.

→ It reduces the speed problem if implemented in hardware.

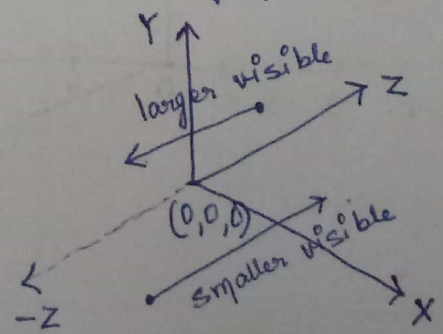
* Disadvantages :-

→ It requires large memory.

→ It is time consuming process.

→ can be applied only to polygon surfaces

* Case - ① :-



When sitting at the +ve z-dir^n, larger value will be visible.

Case-(ii) :-

When sitting at the -ve z-dir^{n}, smaller value will be visible.

→ ALGORITHM :-

Step-(i) :-

Case-(i) :-

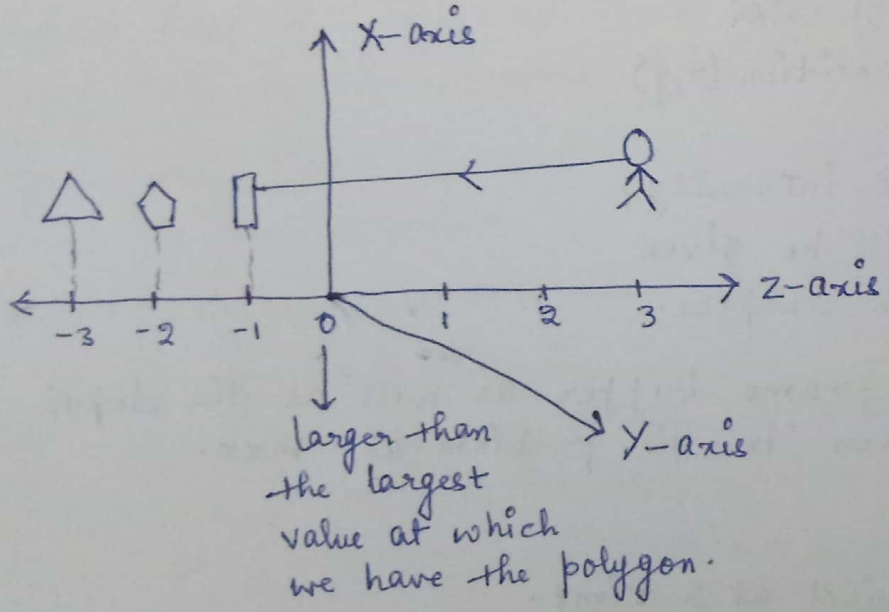
Initialize the depth buffer & refresh buffer.

$d(x,y) = 0$

depth of (x,y) co-ordinate (Initialize it with 0)

$R(x,y) = I_{background}$

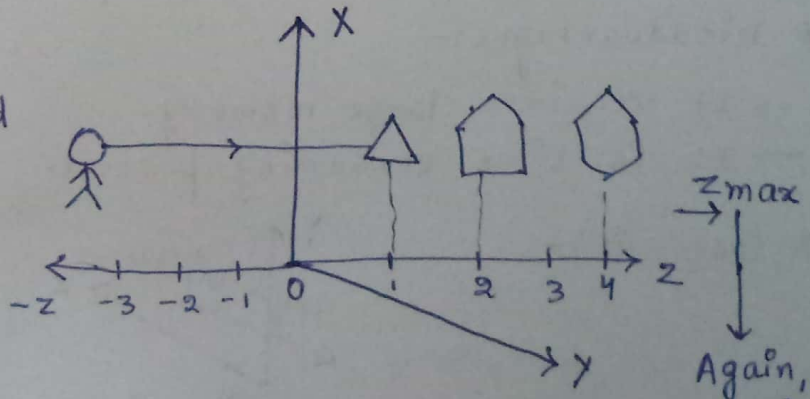
Intensity value / color code will be given for the background.



Case-(ii) :-

$d(x,y) = Z\text{-max}$

$R(x,y) = I_{background}$



Again, Z_{max} is larger than the largest z-value at which we have the polygon.

Step-② :-

We have to check whether $z > d(x,y)$

then, $\boxed{d(x,y) = z}$

↓

update value of $d(x,y)$

And,

$R(x,y) = I(x,y)_{surface}$

↓

Surface for which the $z > d(x,y)$

Repeat Step-② until no more updation is needed for any cell of the frame buffer or z-buffer / depth-buffer

And, the final updated depth-buffer & frame buffer is our solution.

* Note :- Step-2 has been done by assuming case-① as default.

↳ As in case-②,

if $z < d(x,y)$

↳ then, $d(x,y) = z$

& $R(x,y) = I(x,y)_{surface}$

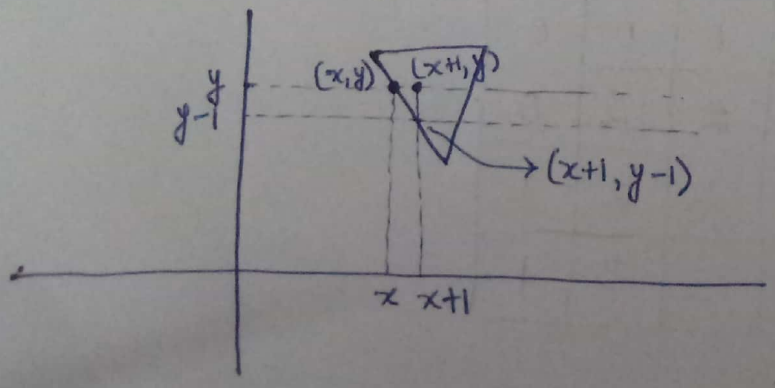
↓

Surface for which $z < d(x,y)$.

* Calculation of the value of z :-

For $Ax + By + Cz + D = 0$

$\Rightarrow z = \dfrac{-Ax - By - D}{C}$



z at (x,y)

z at $(x+1,y)$

z at $(x+1,y-1)$

↳ $A(x+1) + B(y-1) + Cz + D = 0$

↓

obtain z-by solving this.

Q) We have a 4×4 frame buffer & a 4×4 z-buffer / depth buffer.

And, given 3 objects,

Object-①,

x ↓ \ y →	2	3
2	4	4
3	4	4

Color Code → Red (01)

Object-②,

x ↓ \ y →	1	2
1	9	8
2	8	7

Color Code → Green (02)

$I_{background} = 0$

Object-③,

	0	1	2
2	7	6	5
3	7	6	5

Color Code → Blue (03)

A) Depth-Buffer :-

	0	1	2	3
0	0	0	0	0
1	0	~~0~~ 9	~~0~~ 8	0
2	~~0~~ 7	~~0~~ 8	~~0~~ 7	~~0~~ 4
3	~~0~~ 7	~~0~~ 6	~~0~~ 5 4	~~0~~ 4

frame-buffer :-

	0	1	2	3
0	0	0	0	0
1	0	2	2	0
2	3	2	2	1
3	3	3	3	1