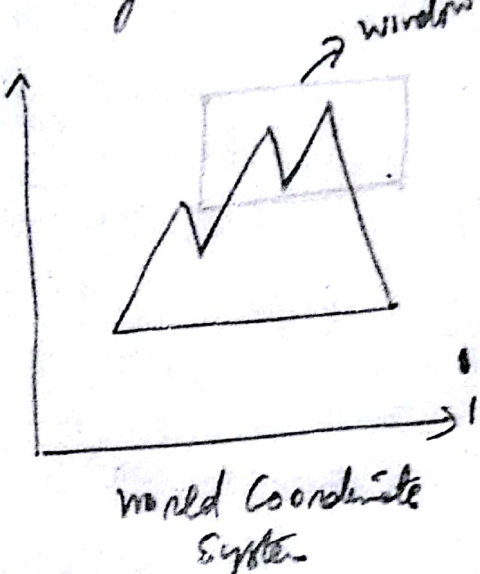
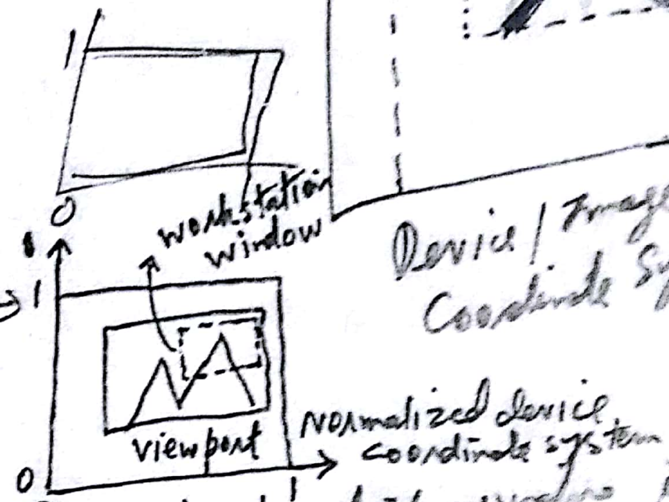
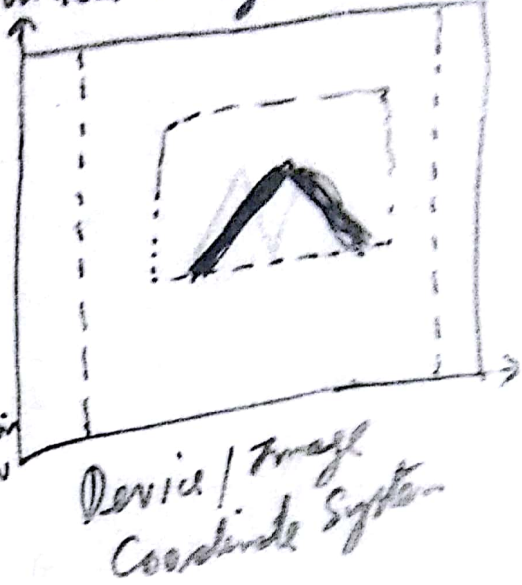
World Coordinate system: Objects are placed in a scene by modelling transformations to a master coordinate system called world coordinate system. (WCS)

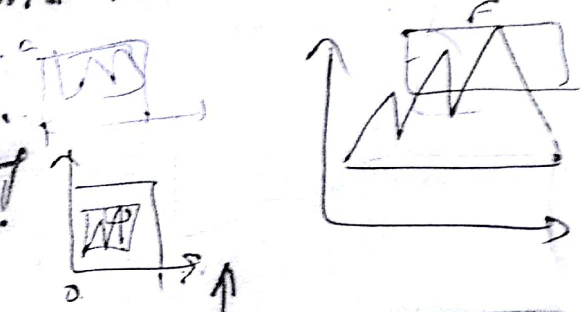
Window: edges parallel to axes of WCS is used to select the portion of scene for which image is to be generated.



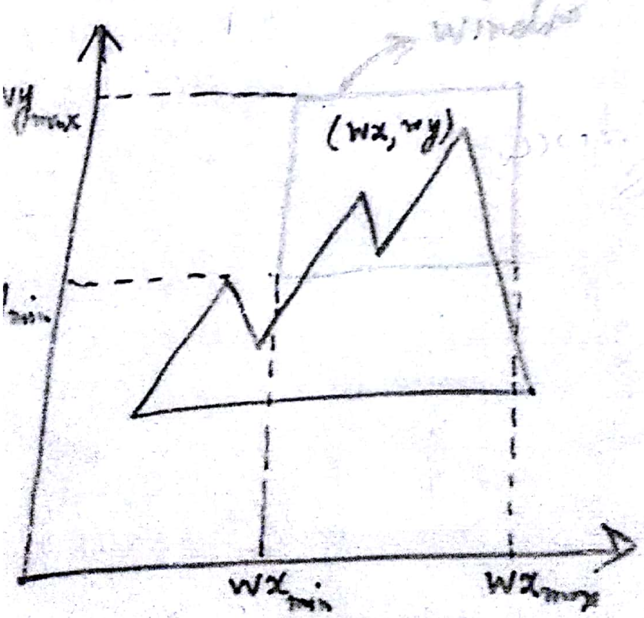


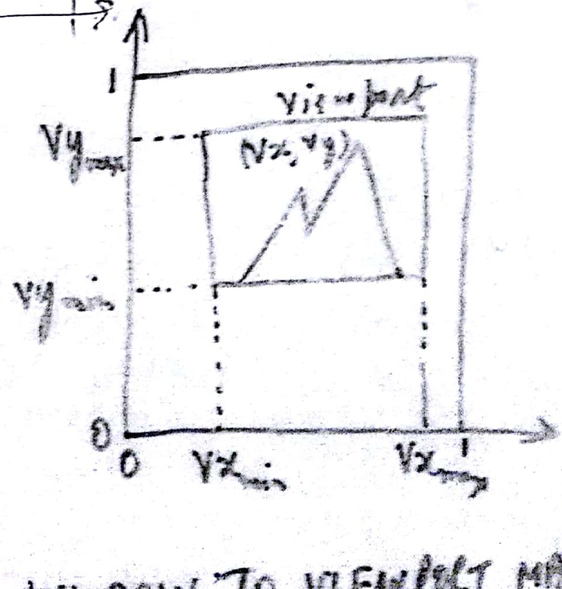


Viewport: area on display device to which window is mapped.



Window to viewport Mapping





WINDOW TO VIEWPORT MAPPING

Objective of window to viewport mapping is to convert world coordinates (wx, wy) of an arbitrary pt to its corresponding normalized device coordinates (vx, vy).

To have same relative placement of pt in viewport as in the window, we have

$$\frac{wx - wx_{min}}{wx_{max} - wx_{min}} = \frac{vx - vx_{min}}{vx_{max} - vx_{min}}$$

Also

$$\frac{wy - wy_{min}}{wy_{max} - wy_{min}} = \frac{vy - vy_{min}}{vy_{max} - vy_{min}}$$

$$\therefore \quad vx = \frac{vx_{max} - vx_{min}}{wx_{max} - wx_{min}}(wx - wx_{min}) + vx_{min}$$

$$vy = \frac{vy_{max} - vy_{min}}{wy_{max} - wy_{min}}(wy - wy_{min}) + vy_{min}$$

Since all eight coords of window & viewport are consts

$$\therefore \begin{pmatrix} vx \\ vy \\ 1 \end{pmatrix} = N \cdot \begin{pmatrix} wx \\ wy \\ 1 \end{pmatrix}$$

where

$$N = \begin{pmatrix} 1 & 0 & vx_{min} \\ 0 & 1 & vy_{min} \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \frac{vx_{max} - vx_{min}}{wx_{max} - wx_{min}} & 0 & 0 \\ 0 & \frac{vy_{max} - vy_{min}}{wy_{max} - wy_{min}} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & -Wx_{min} \\ 0 & 1 & -Wy_{min} \\ 0 & 0 & 1 \end{pmatrix}$$

$x_{min} \leq x \leq x$.

&

POINT CLIPPING

$x_{min} \leq x \leq x_{max}$ & $y_{min} \leq y \leq y_{max}$

where x_{min}, x_{max}, y_{min}, y_{max} is the window.

A pt (x, y) is inside the window if inequalities are true.

LINE CLIPPING

Cohen- Sutherland Algo (Line clipping Algo).

Algo divided into 2 phases.

1. Identify lines which intersect clipping window & so need to be clipped

(2) Perform the clipping.

All lines fall in following categories

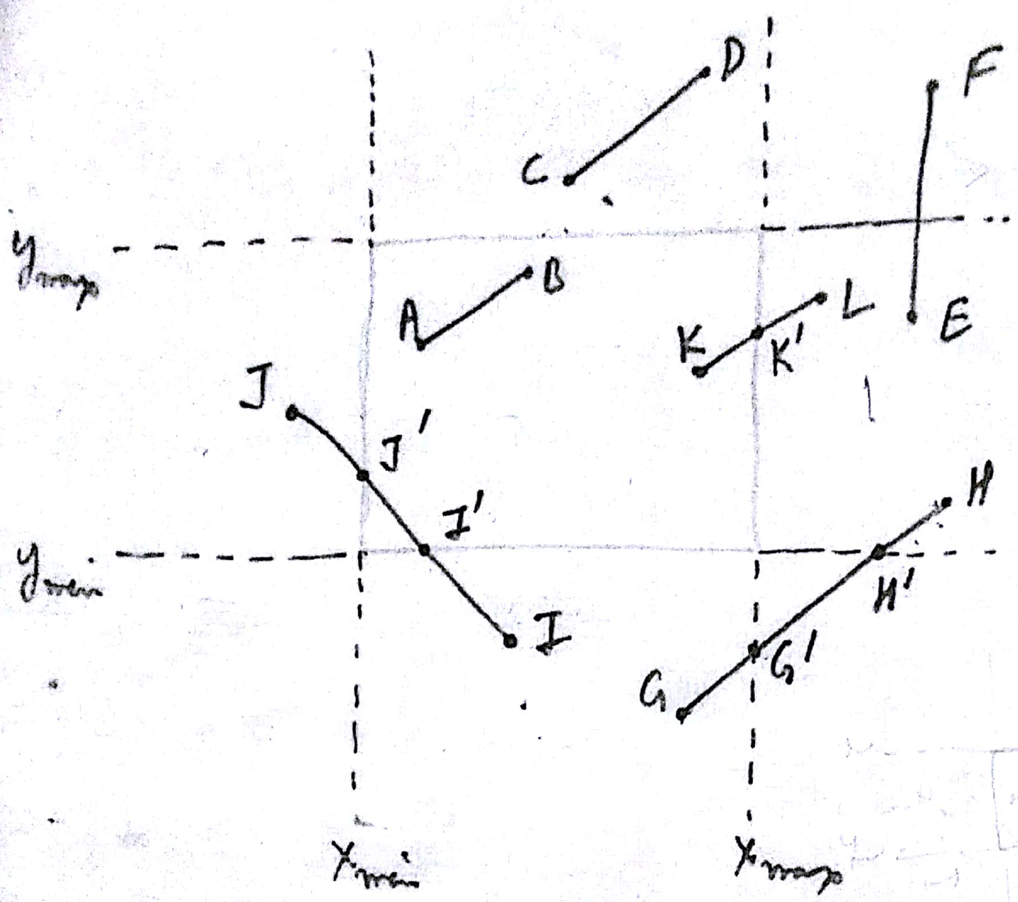
1. Visible - both end pts of line lie in the window.
2. Not visible - line definitely lies outside the window

This occurs if line from (x_1, y_1) to (x_2, y_2) satisfies any one of the following

$x_1, x_2 > x_{max}$ $\quad$ $y_1, y_2 > y_{max}$

$x_1, x_2 < x_{min}$ $\quad$ $y_1, y_2 < y_{min}$

Clipping Candidate – line is neither category 1 nor 2.



Procedure

1. Assign a 4 bit region code to each endpt of line. Code determines according to which of the following 9 regions of the plane the end point lies in.

		x_{min}		x_{max}		
		1001	1000	1010		T
y_{max}	L	0001	0000	0010		R
y_{min}		0101	0100	0110		B

T B R L

	T	
1001	1000	1010
L 0001	0000	0010 R
0101	0100	0110
	B	

TBRL

Starting from leftmost bit, each bit of code is set to true(1) or false(0) according to following

Bit 1 = endpt is above the window = sign $(y - y_{max})$

Bit 2 = " " " below " " = sign $(y_{min} - y)$

Bit 3 = " " " right of the " " = sign $(x - x_{max})$

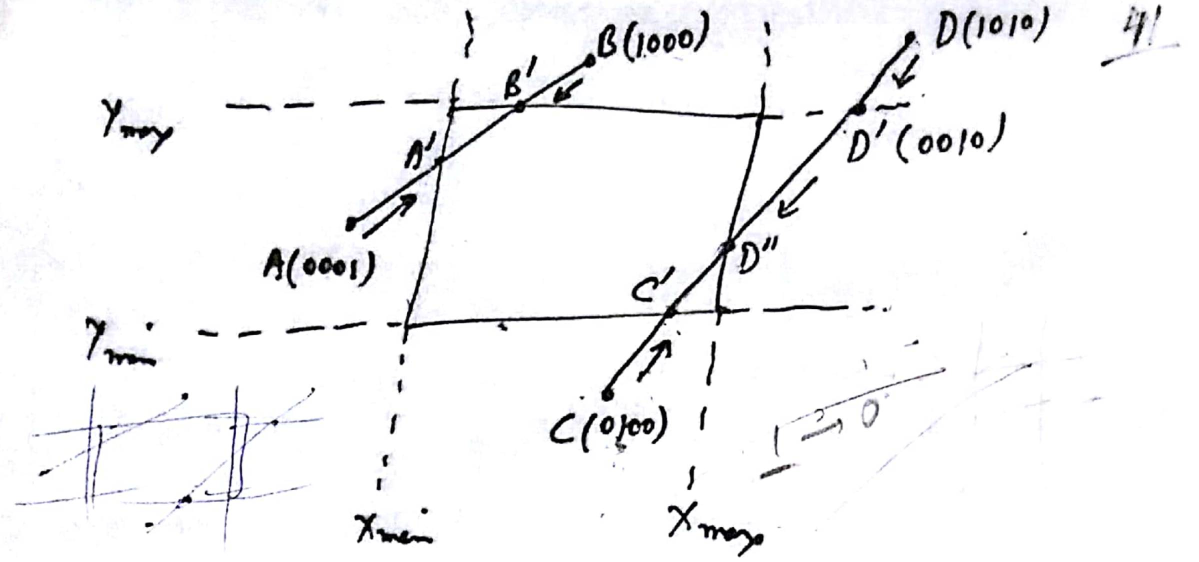
Bit 4 = " " " left " " " = sign $(x_{min} - x)$.

Conventions followed: sign(a) = 1 if a is +ive, otherwise zero.

pt with code 0000 is inside the window.

2. Line is visible if both region codes are 0000 & not visible if bitwise logical AND of the code is not 0000 & a candidate for clipping if bitwise AND of the region code is 0000.

For category 3 line, find intersection pt of line with boundaries of clipping window or infinite extension of one of the boundaries.



Let (x_1, y_1) be endpt of line which is outside the window i.e whose region code is not 0000. Select extended boundary line by observing that those boundary lines that are candidates for intersection are the ones which for which the chosen endpt must be pushed across so as to change a 1 in its code to 0. i.e

If bit 1 is 1, intersect with line $y = y_{max}$
" " 2 " ", " " " $y = y_{min}$
" " 3 " ", " " " $x = x_{max}$
" " 4 " 1, " " " $x = x_{min}$

If endpt C is chosen, then bottom boundary line $y = y_{min}$ is selected for intersection. If D is chosen as endpt then either $y = y_{max}$ or $x = x_{max}$ is used. Coords of intersection are D'' $x_i = x_{min}$ or x_{max} if boundary line is vert

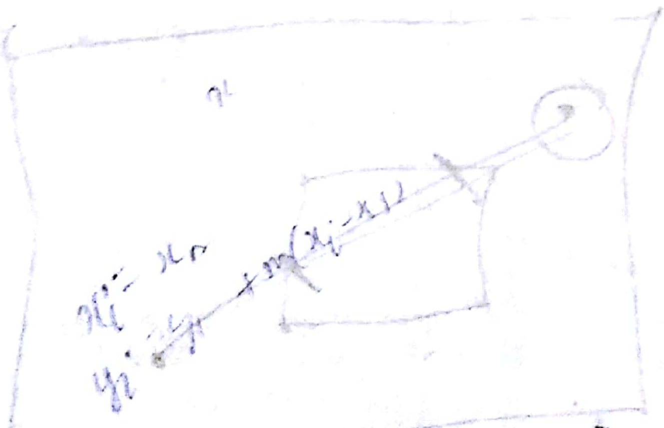
$$y_i = y_1 + m(x_i - x_1)$$

or $x_i = x_1 + (y_i - y_1)/m$ if boundary line is horizontal

$y_i = y_{min}$ or y_{max}

where $m = (y_2 - y_1)/(x_2 - x_1)$

We replace end pt (x_1, y_1) with intersection pt (x_i, y_i) eliminating portion of original line outside the window. Now new endpt is assigned an updated region code & clipped line re-categorized & handled in same way. This iterative process terminates when we finally reach clipped line that belongs to visible or not visible category.



$y_i = y_1 + m(x_i$

1001	1000	1010
0001	0000	0010
0101	0100	0110

L R

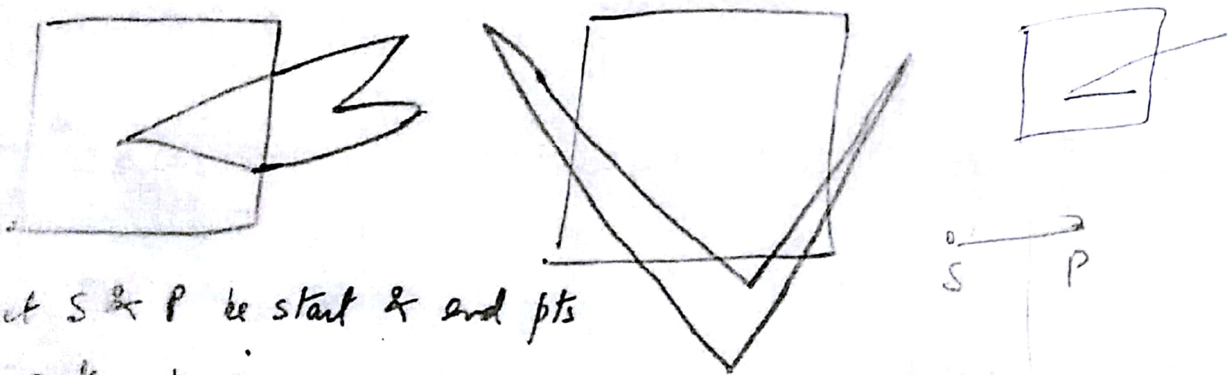
bit 1 : left
(Right bit)

bit 2 : right

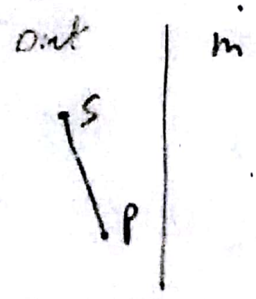
bit 3 : below

bit 4 : above.

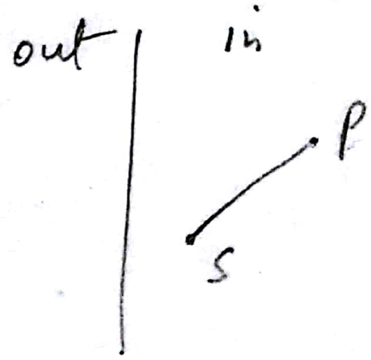
POLYGON CLIPPING (SUTHERLAND HODGMAN

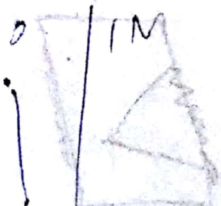


Let S & P be start & end pts

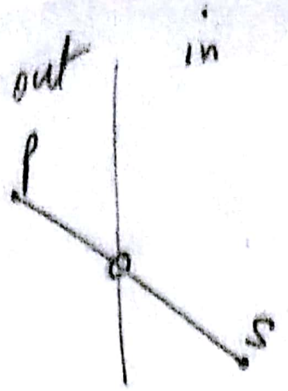


S & P outside
No output

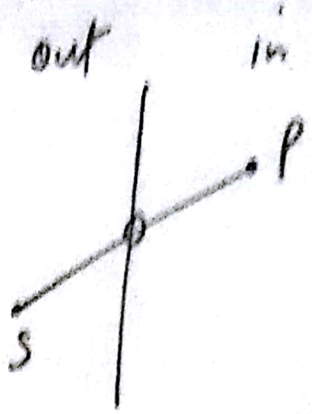




S & P inside
Output P

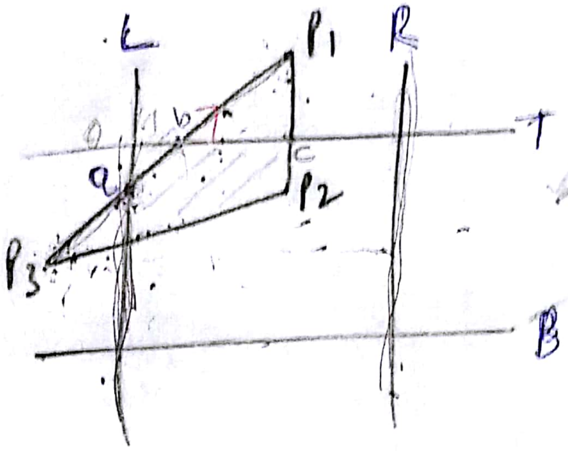


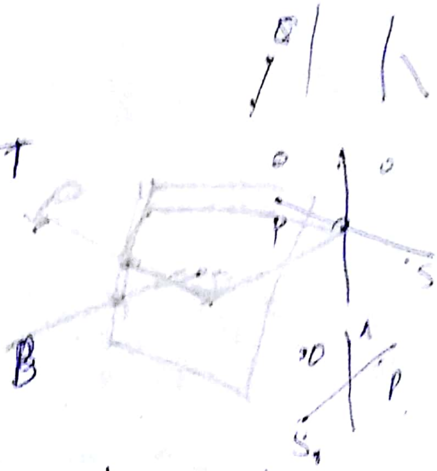
S & P inside & outside
Output is intersection



S outside & P inside
Output is intersection & P

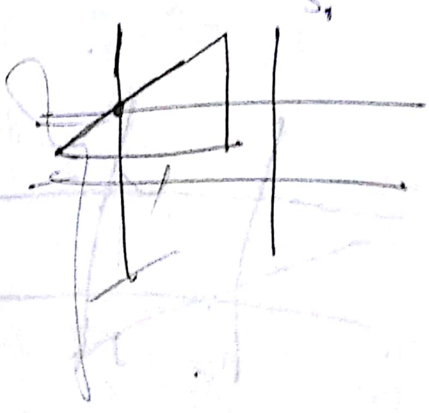
SITUATION	OUTPUT
S outside → P outside	None
S inside → P inside	P
S inside → P outside	Intersection
S outside → P inside	Intersection plus P





P3. L P3 R P3 B P3 T
P1 a.
P1

P3 L P3. R. P3 B P3 T
P1 a. a B T
P1 R



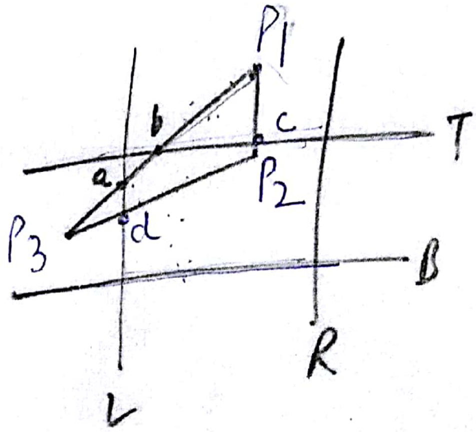
P3 P3 P3 P3
P1 L a R a B a T
P1 R B T

P3 P3 P3 P3
L a R a B a T a
P1 P1 R B T

P3 P3 P3 P3 T a
L a R a B a T
P1 P1 R P1 B

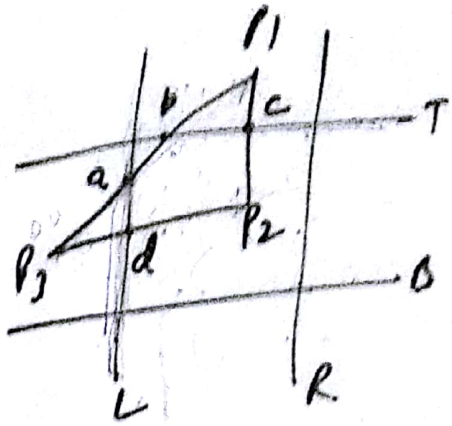
P3 P3 P3 P3 T a
L a R a B a T
P1 P1 R P1 B P1 T

P3 P3 P3 P3 T a
L a R a B a T b
P1 P1 R P1 B P1 T



$P_2 \quad L \; \frac{P_1}{P_2} \quad R \; \frac{P_1}{P_2} \quad B \; \frac{P_1}{P_2} \quad T \; \frac{c}{P_2}$

$\frac{P_2}{P_3} \quad L \; \frac{P_2}{d} \quad R \; \frac{P_2}{d} \quad B \; \frac{P_2}{d} \quad T \; d$



Summary

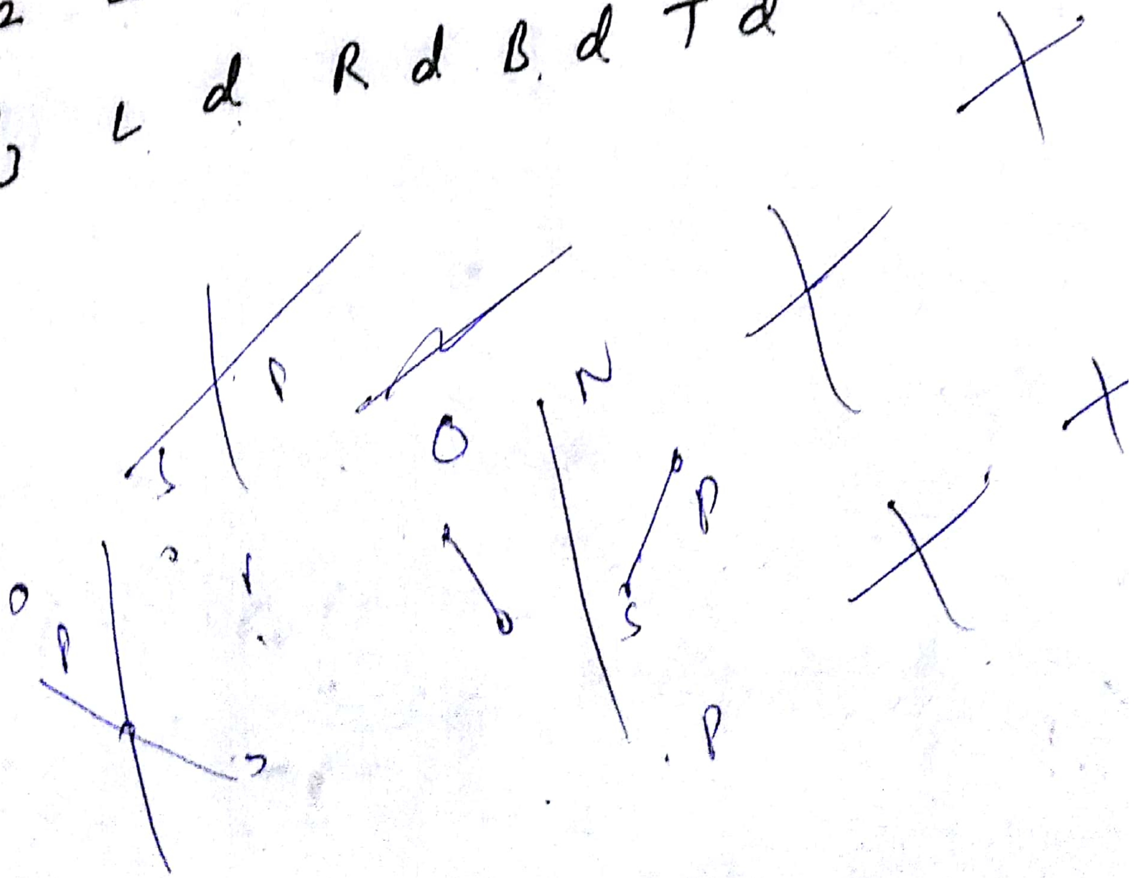
$P_3 \quad L \; \frac{P_3}{a} \quad R \; \frac{P_3}{a} \quad B \; \frac{P_3}{a} \quad T \; a$

$P_1 \quad L \; \frac{a}{P_1} \quad R \; P_1 \quad B \; P_1 \quad T \; b$

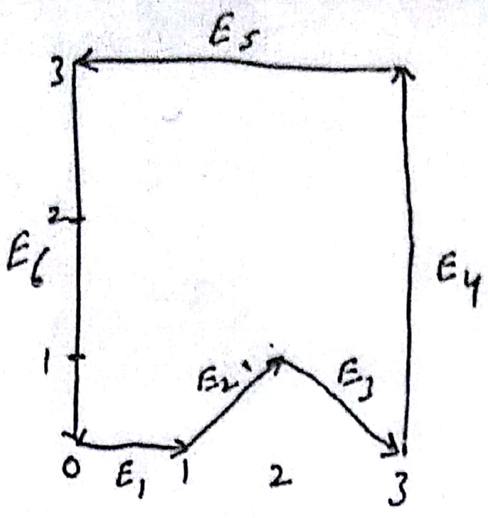
$P_2 \quad L \; P_2 \quad R \; P_2 \quad B \; P_2 \quad T \; \frac{c}{P_2}$

$P_2 \quad L \; P_2 \quad R \; d \quad B \; d \quad T \; d$

$P_3 \quad L \; d$



46 A



$E_1 = (1,0,0)$ $E_2 = (1,1,0)$

$E_3 = (1,-1,0)$ $E_4 = (0,3,0)$

$E_5 = (-3,0,0)$ $E_6 = (0,-3,0)$

z component is zero, since all edges in xy plane.

Cross product $E_i \times E_j$ for 2 successive edges is a vector perpendicular to xy plane with z component equal to $E_{ix} E_{jy} - E_{jx} E_{iy}$

$E_1 \times E_2 = (0,0,\ )$ $E_2 \times E_3 = (0,0,\ )$

$E_3 \times E_4 = (0,0,\ )$ $E_4 \times E_5 = (0,0,\ )$

$E_5 \times E_6 = (0,0,\ )$ $E_6 \times E_1 = (0,0,\ )$

If cross product is negative and positive then polygons are concave

Convex polygons are correctly clipped by Sutherland Hodgeman