Midterm Exam

300 points

290

45 **1. True or False and CORRECT when false [15 points each]:**

15 a. Lighting is performed before the viewing transformation in the 3D rendering pipeline.

True

15 b. Clipping using image-based techniques is faster than object-based techniques since it avoids computations in 3D space. slower

it draws many hidden things

False.

15 c. In parallel projection, the direction of projection (DOP) is the same for all points and parallel to the viewing direction as well as the view plane.

False and Perpendicular to the view plane

50 **2. Complete the following:**

10 a. [10 points] Addition of a point and a vector = ...new Point.... while addition of two points = ...meaningless (nothing)....

10 b. [10 points] Viewport transformation converts from....projection's 2D.... coordinate to....Device's 2D.... coordinate.

10 c. [10 points] Clipping planes are specified by....zNear.... and ....zFar....

5 d. [10 points] Parallel projection has two types; ....orthogonal.... and ....

15 e. [15 points] In the camera model, the camera is defined by

i. ....Camera Position....

ii. ....Look vector....

iii. ....up vector....

5 3. [10 points] What caused vector displays to be favored in the 1970s compared to raster displays?

- no need for very high resolution
- Cost
memory

15 4. [15 points] Mention three ways to find out if a polygon is concave or not?

- vector method
- rotational method
- (The line joining any two points is not completely inside the Polygon)

40 5. In Cohen-Sutherland line clipping algorithms, mention the condition checked on the codes of the two endpoints for:

20 a. [20 points] Trivial accept

OR operation of the two endpoint is equal to 0000

20 b. [20 points] Trivial reject

And operation of the two endpoint is not equal to 0000

$$\begin{bmatrix} 1/s_{x_1} & 0 & 0 \\ 0 & 1/s_{y_1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_{x_2} & 0 & 0 \\ 0 & s_{y_2} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_{x_2}/s_{x_1} & 0 & 0 \\ 0 & s_{y_2}/s_{y_1} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

25 6. [25 points] Simplify the transformations $\left(R(\theta)S(s_{x1}, s_{y1})T(t)\right)^{-1} R(\theta)S(s_{x2}, s_{y2})T(t)$

$= T^{-1}(t)\, S^{-1}(s_{x_1}, s_{y_1})\, R^{-1}(\theta)\, R\theta\, S(s_{x_2}, s_{y_2})\, T(t)$

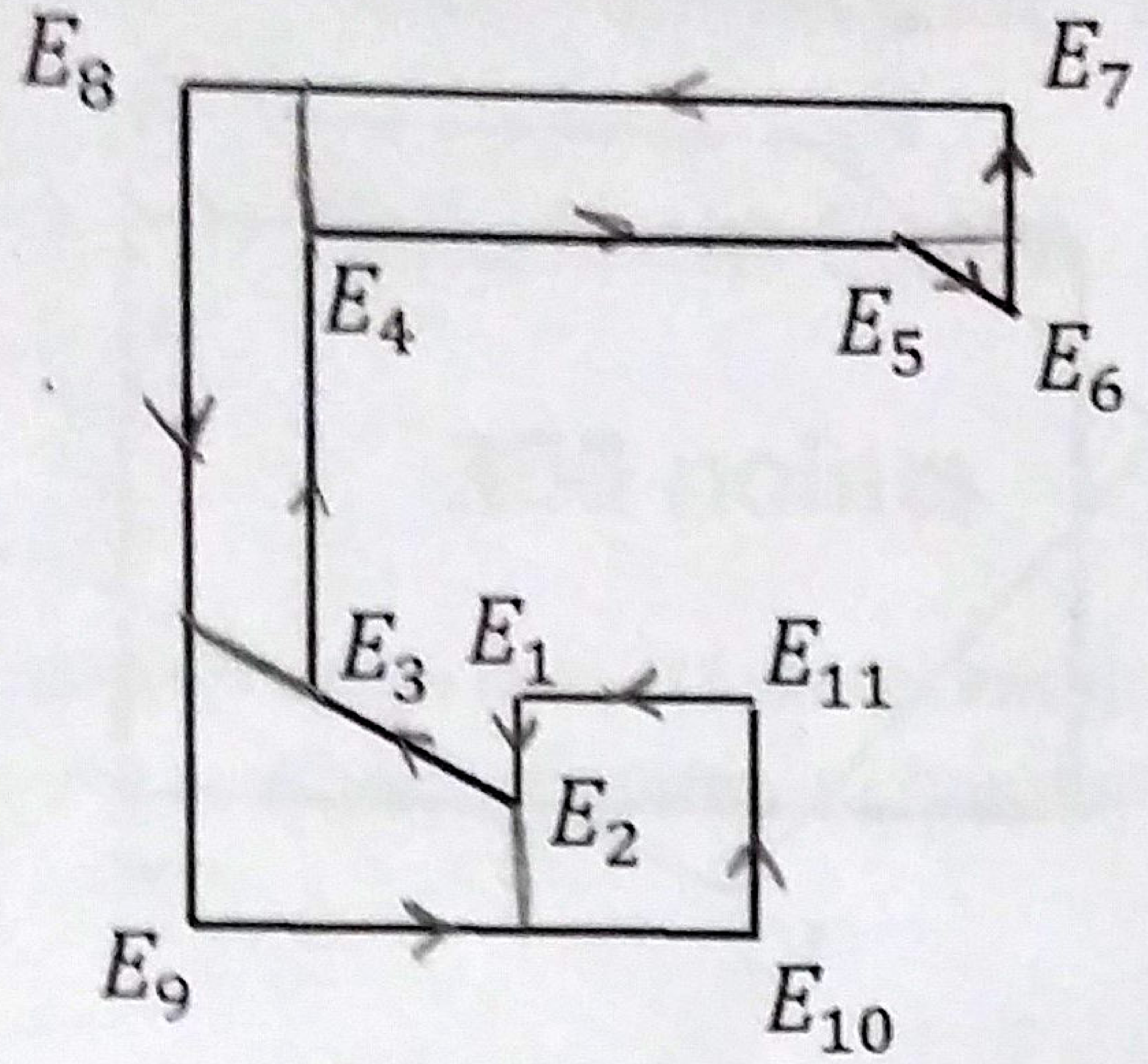
$= T(-t)\, S(1/s_{x_1}, 1/s_{y_1})\, R(-\theta)\, R\theta\, S(s_{x_2}, s_{y_2})\, T(t)$

$= T(-t)\, S(1/s_{x_1}, 1/s_{y_1})\, S(s_{x_2}, s_{y_2})\, T(t)$

$= T(-t)\, S\left(\frac{s_{x_2}}{s_{x_1}}, \frac{s_{y_2}}{s_{y_1}}\right) T(t)$

50

20 7. [20 points] Apply the vector method to split the following polygon. You do not need to show intermediate steps, just show the lines splitting the polygon based on processing the vertices in their order shown below.



30 8. [30 points] Use Bresenham's line drawing algorithm to render the line between the two endpoints P1=(-2,4) and P2=(1,-3).

ofc → offset scaled vector = 2dx - dy

dy = -7 , dx = 3

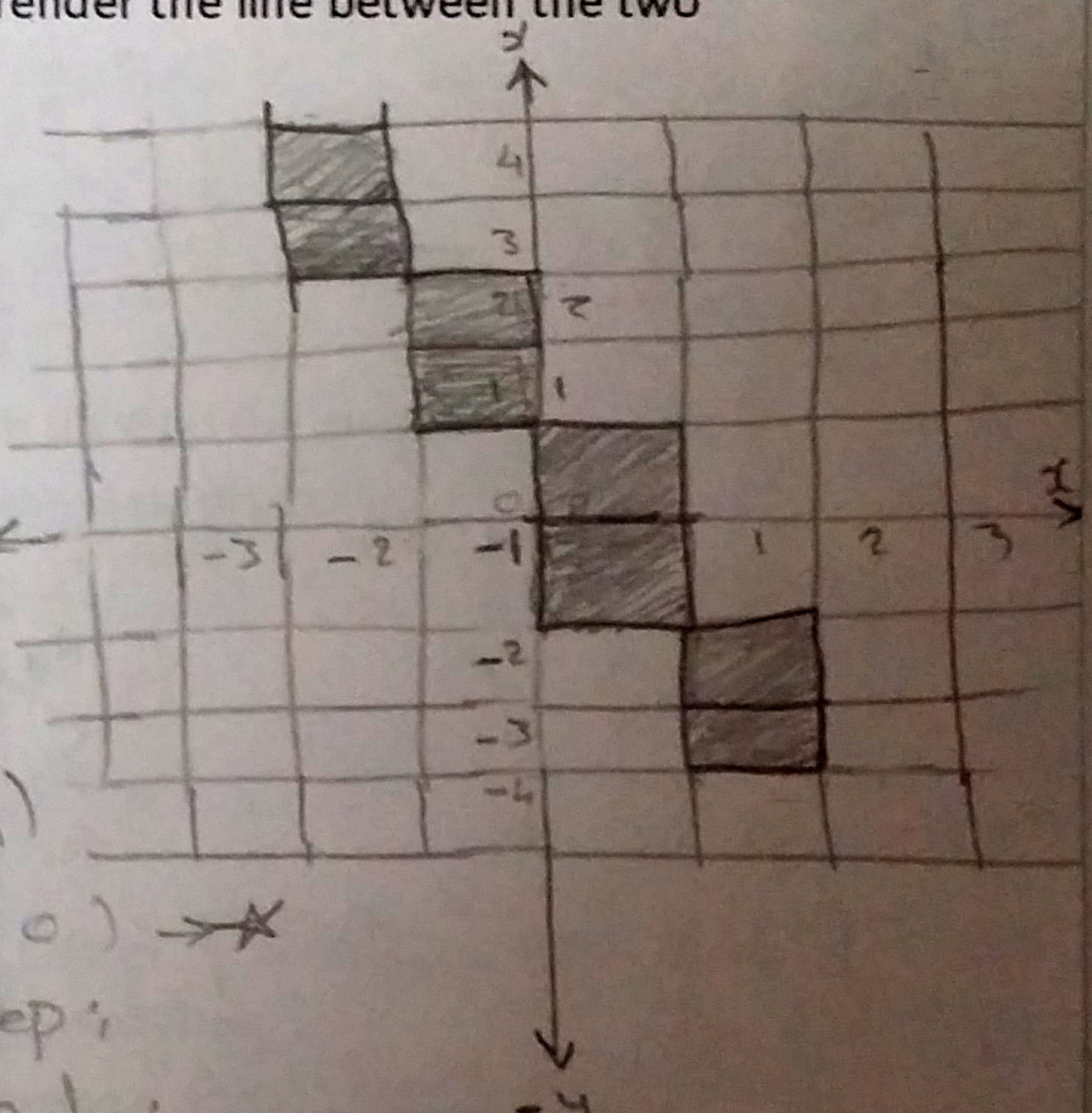
abs(dy) = 7

abs(dx) < abs(dy)

∴ slope > 1

$y_2 < y_1$ → ystep = -1

$x_2 > x_1$ → xstep = 1



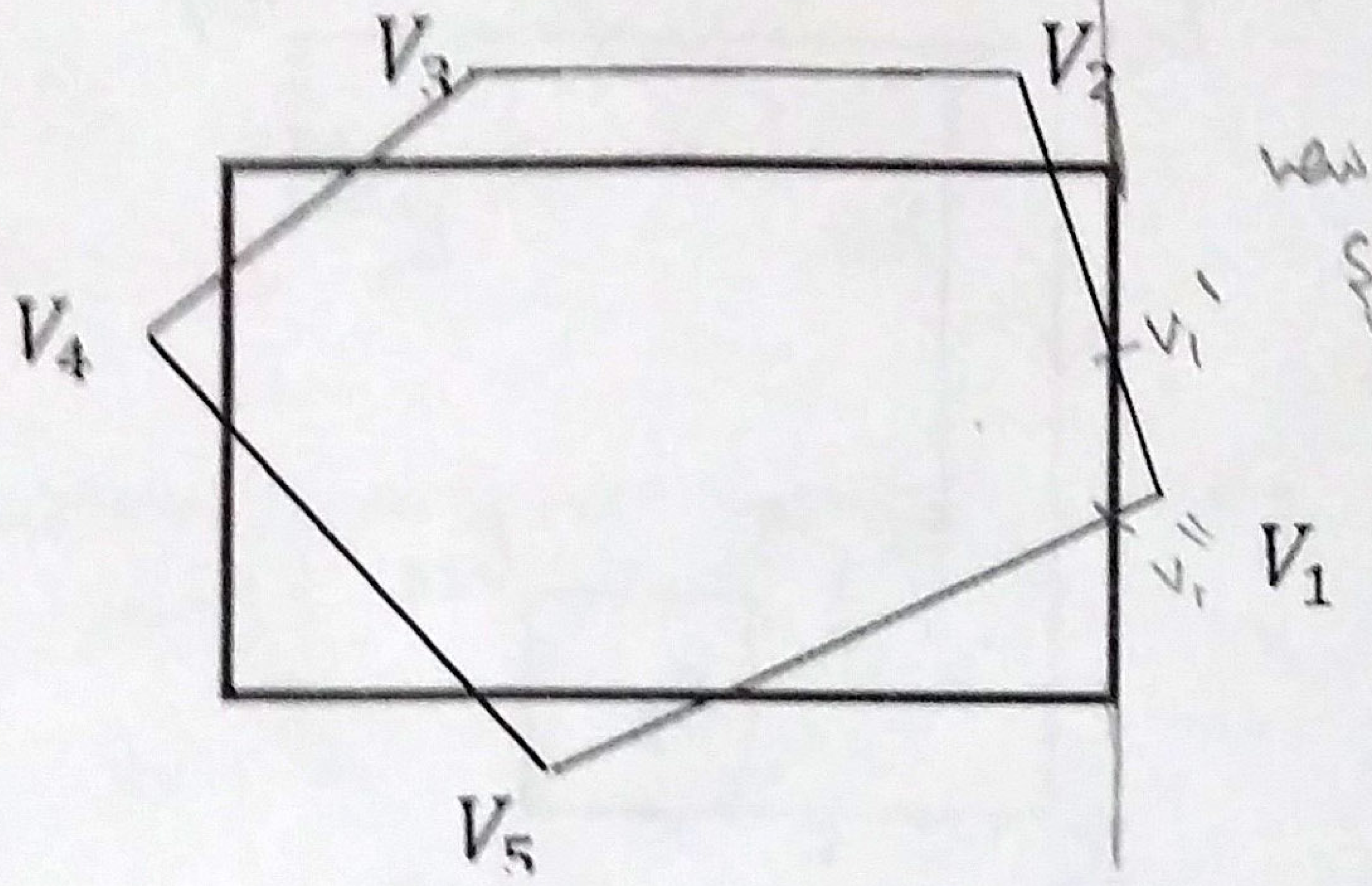
y	x	ofc
4	-2	-1 → 5
3	-2	* 5 → -3
2	-1	-3 → 3
1	-1	* 3 → -5
0	0	-5 → 1
-1	0	* 1 → -7
-2	1	-7 → -1
-3	1	-1 → 5

```
while(y0 != y1)
{   if(ofc >= 0) → *
    { x += xstep;
      ofc -= 2dy;
    }
    ofc += 2dx
    y+=ystep
    draw(x,y);
}
```

rules:
* 1st out/ 2nd in → move 1st to boundry and accept 1st' and 2nd
* 1st in/2nd in → accept second
* 1st in /2nd out → move 2nd to boundry & accept second'
* 1st out /2nd out → reject both

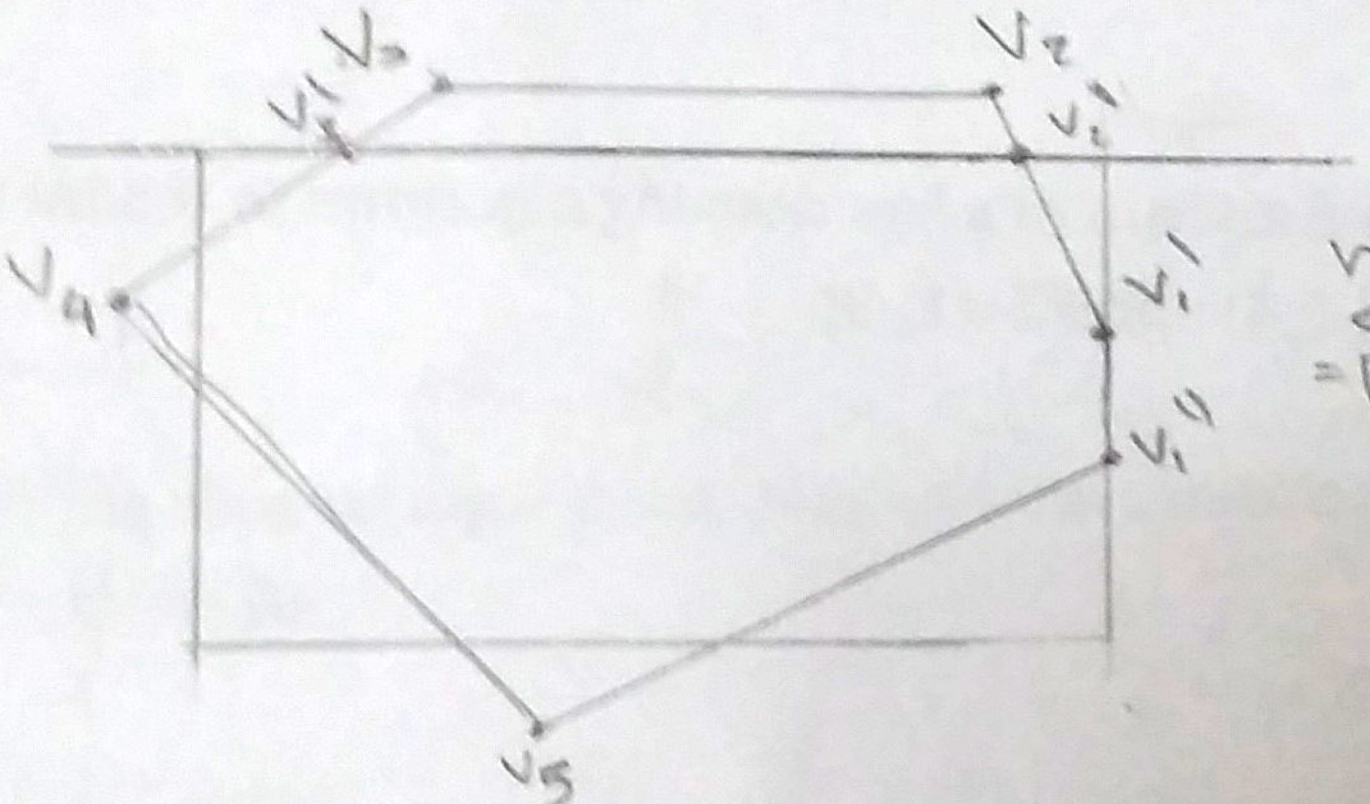
60

9. [60 points] Apply Sutherland-Hodgman on the figure in the order of boundaries (right, top, left, bottom):



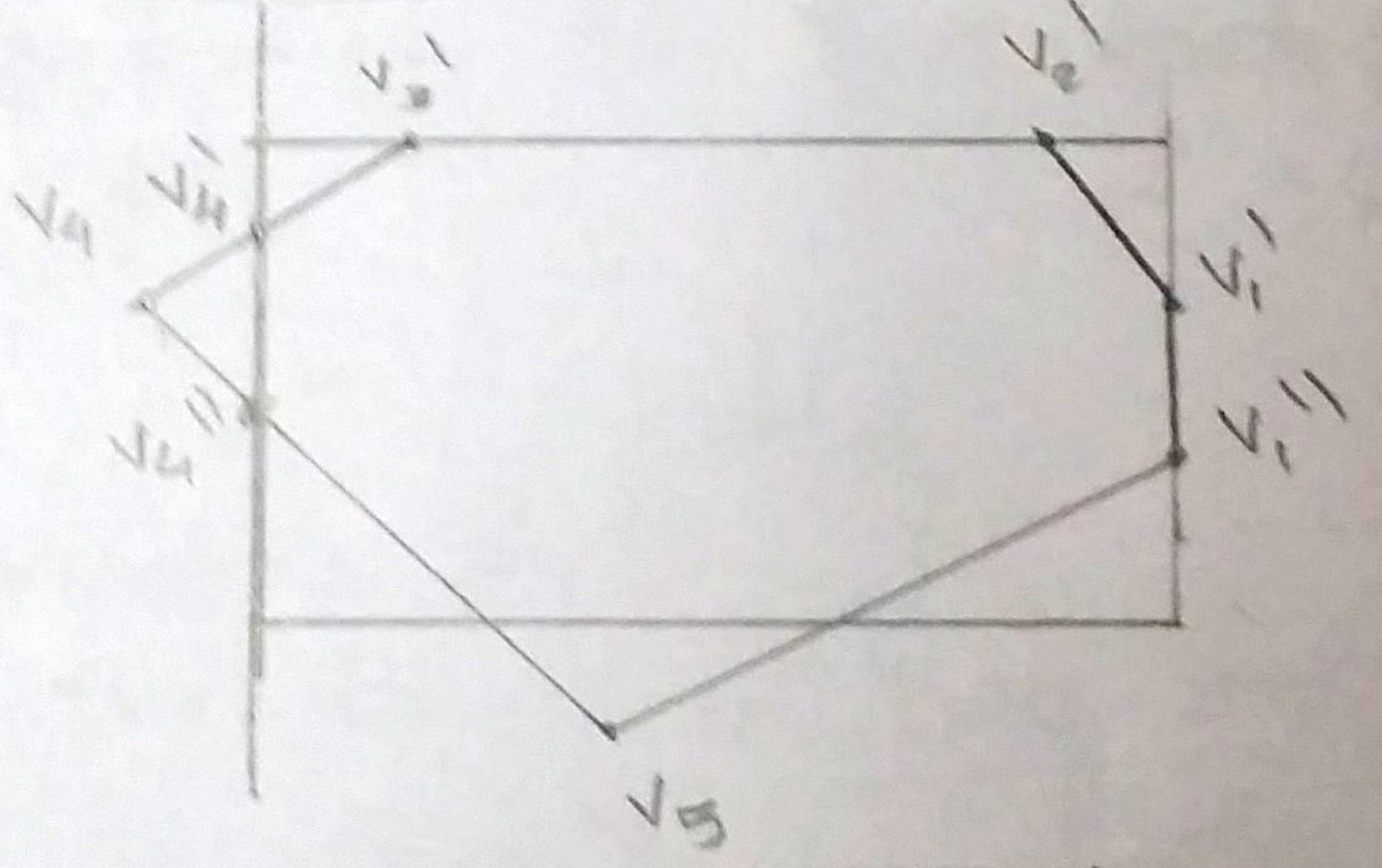
$V_1, V_2 \to V_1', V_2$
$V_2, V_3 \to V_3$
$V_3, V_4 \to V_4$
$V_4, V_5 \to V_5$
$V_5, V_1 \to V_1''$

new list = $\{V_1', V_2, V_3, V_4, V_5, V_1''\}$



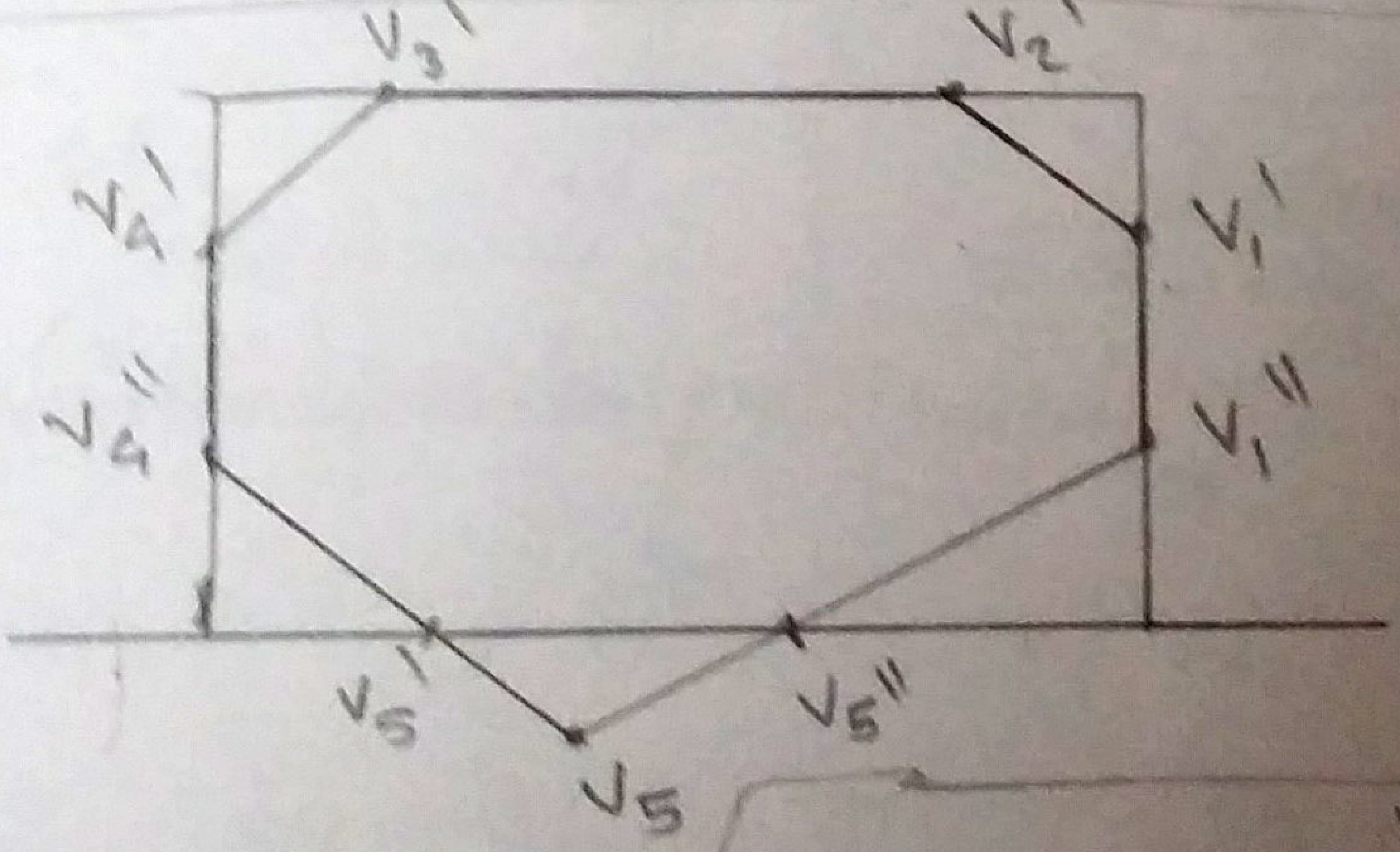
$V_1', V_2 \to V_2'$
$V_2, V_3 \to$ none
$V_3, V_4 \to V_3', V_4$
$V_4, V_5 \to V_5$
$V_5, V_1'' \to V_1''$
$V_1'', V_1' \to V_1'$

new list = $\{V_2', V_3', V_4, V_5, V_1'', V_1'\}$



$V_2', V_3' \to V_3'$
$V_3', V_4 \to V_4'$
$V_4, V_5 \to V_4'', V_5$
$V_5, V_1'' \to V_1''$
$V_1'', V_1' \to V_1'$
$V_1', V_2 \to V_2$

new list 3 $\{V_3', V_4', V_4'', V_5, V_1'', V_1', V_2'\}$



$V_3', V_4' \to V_4'$
$V_4', V_4'' \to V_4''$
$V_4'', V_5 \to V_5'$
$V_5, V_1'' \to V_5'', V_1''$
$V_1'', V_1' \to V_1'$
$V_1', V_2' \to V_2'$
$V_2', V_3' \to V_3'$

Final list:
$\{V_4', V_4'', V_5', V_5'', V_1'', V_1', V_2', V_3'\}$