Computer Graphics

Unit 3 – Part 2

-By Manjula. S

Clipping Window

- The mapping of a two-dimensional, world-coordinate scene description to device coordinates is called a **two-dimensional viewing transformation.**
- This transformation is simply referred to as the *window-to-viewport transformation* or the *windowing transformation.*
- In analogy with three-dimensional viewing, we can describe the steps for two-dimensional viewing as indicated in Figure 2.

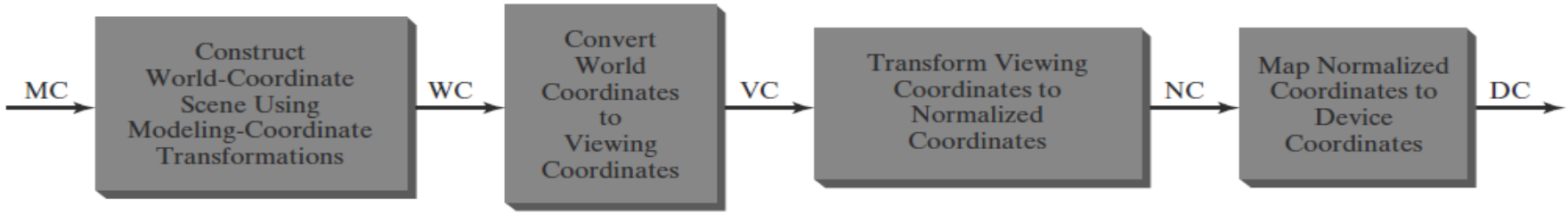


FIGURE 2
Two-dimensional viewing-transformation pipeline.

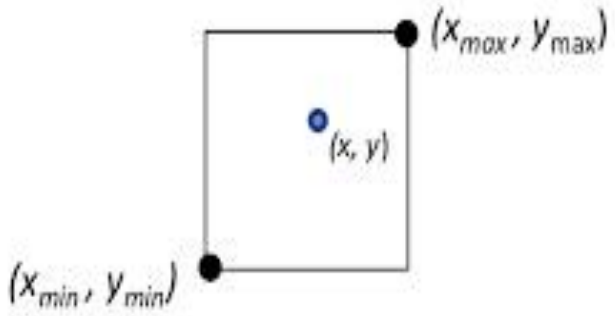
Clipping Types

Clipping Types:

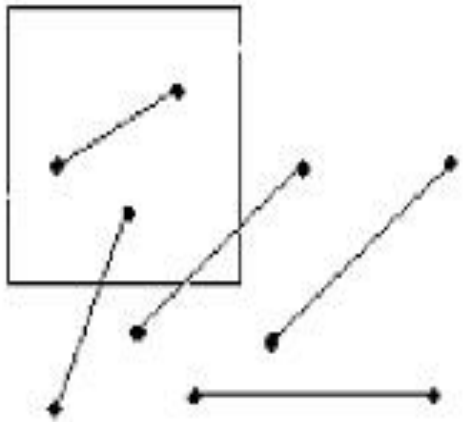
- Point Clipping
- Line Clipping
- Polygon Clipping(Fill area Clipping)
- Curve Clipping
- Text Clipping

Line Clipping

- Clipping a point (x, y)
  - if ($x_{min} < x < x_{max}$) && ($y_{min} < y < y_{max}$)
  - Then: (x, y) is inside



- For lines:
  - If both endpoints are in, do "trivial acceptance"
  - If one endpoint in, one endpoint out, must clip
  - If both endpoints out:
    - Could be out
    - Could be clipped



- Brute force clip: solve simultaneous equations using $y = mx + b$ for the line, and the four edges (of the region)
  - Slope-intercept formula handles infinite line only
  - Doesn't handle vertical lines

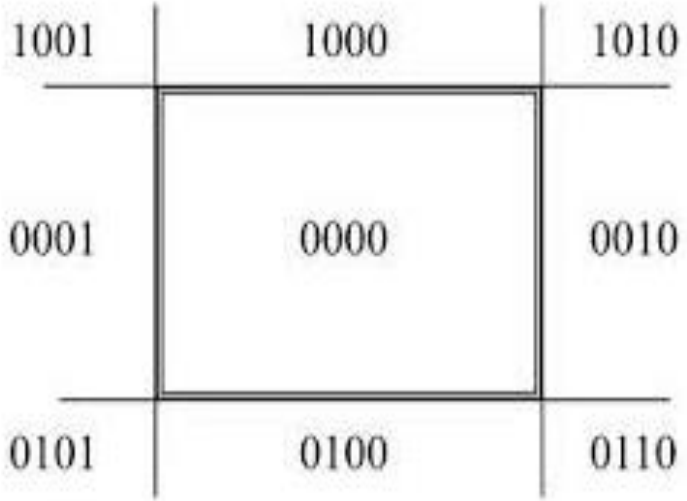
-By Manjula. S

Cohen-Sutherland line clipping: steps

- Calculate differences between endpoint coordinates and clipping boundaries
- Use the resultant sign bit of each difference to set the corresponding value in the region code
  - Bit 1 is the sign bit of $x-xw_{min}$
  - Bit 2 is the sign bit of $xw_{max}-x$
  - Bit 3 is the sign bit of $y-yw_{min}$
  - Bit 4 is the sign bit of $yw_{max}-y$
- Any lines that are completely inside have a region code 0000 for both endpoints (save the line segment)
- Any line that has a region code value of 1 in the same bit position for each endpoint is completely outside (eliminate the line segment)

Cohen-Sutherland Clipping (cont.)

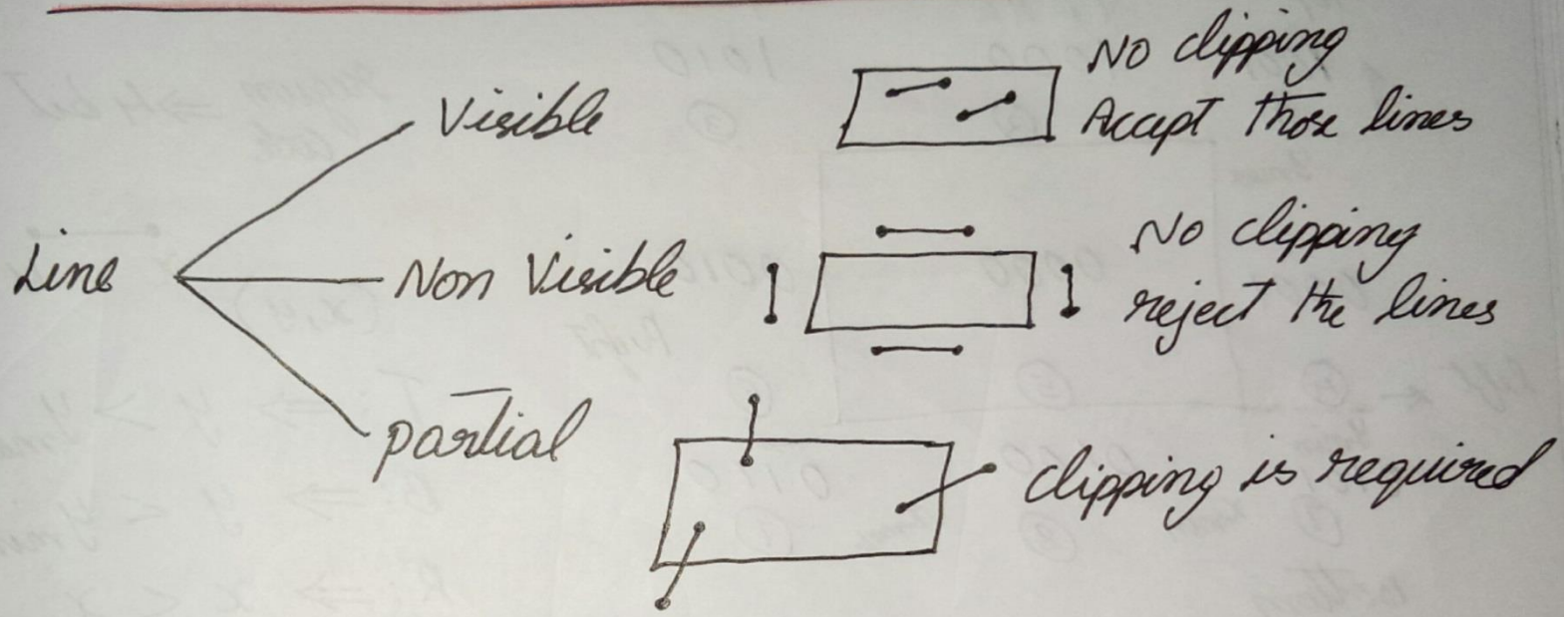
Bit 1: Above
Bit 2: Below
Bit 3: Right
Bit 4: Left



- A line can be trivially accepted if both endpoints have an outcode of 0000.
- A line can be trivially rejected if any corresponding bits in the two outcodes are both equal to 1. (This means that both endpoints are to the right, to the left, above, or below the window.)
- if (outcode 1 & outcode 2) != 0000, trivially reject!

-By Manjula. S

Cohen Sutherland Line clipping



code	code	code
region (1)	region (2)	region (3)
code	code	code
region (4)	region (5)	region (6)
code	code	code
region (7)	region (8)	region (9)

code → 4 bits

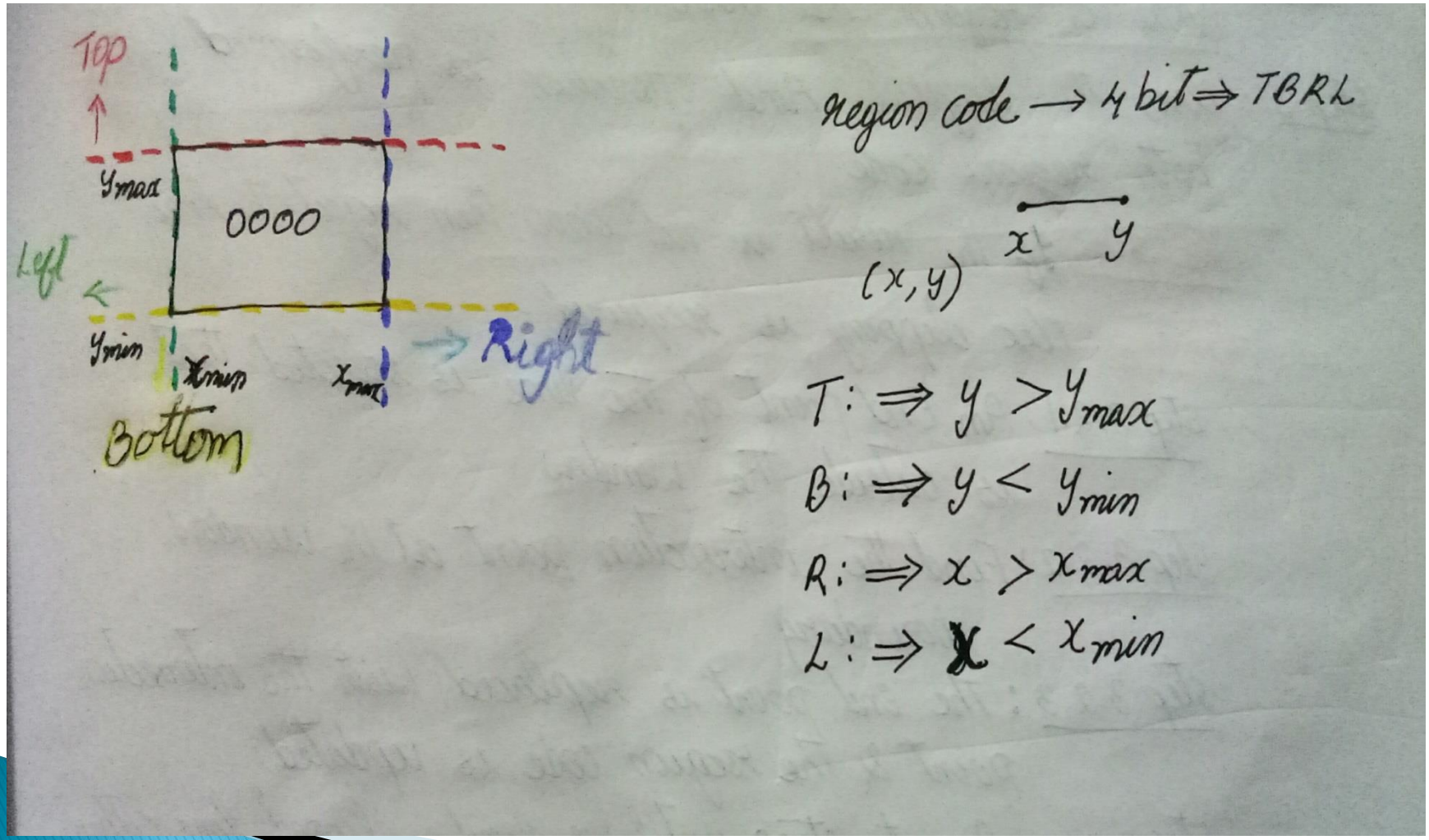
0 1

4 bits ⇒ T B R L

Top Bottom Right Left

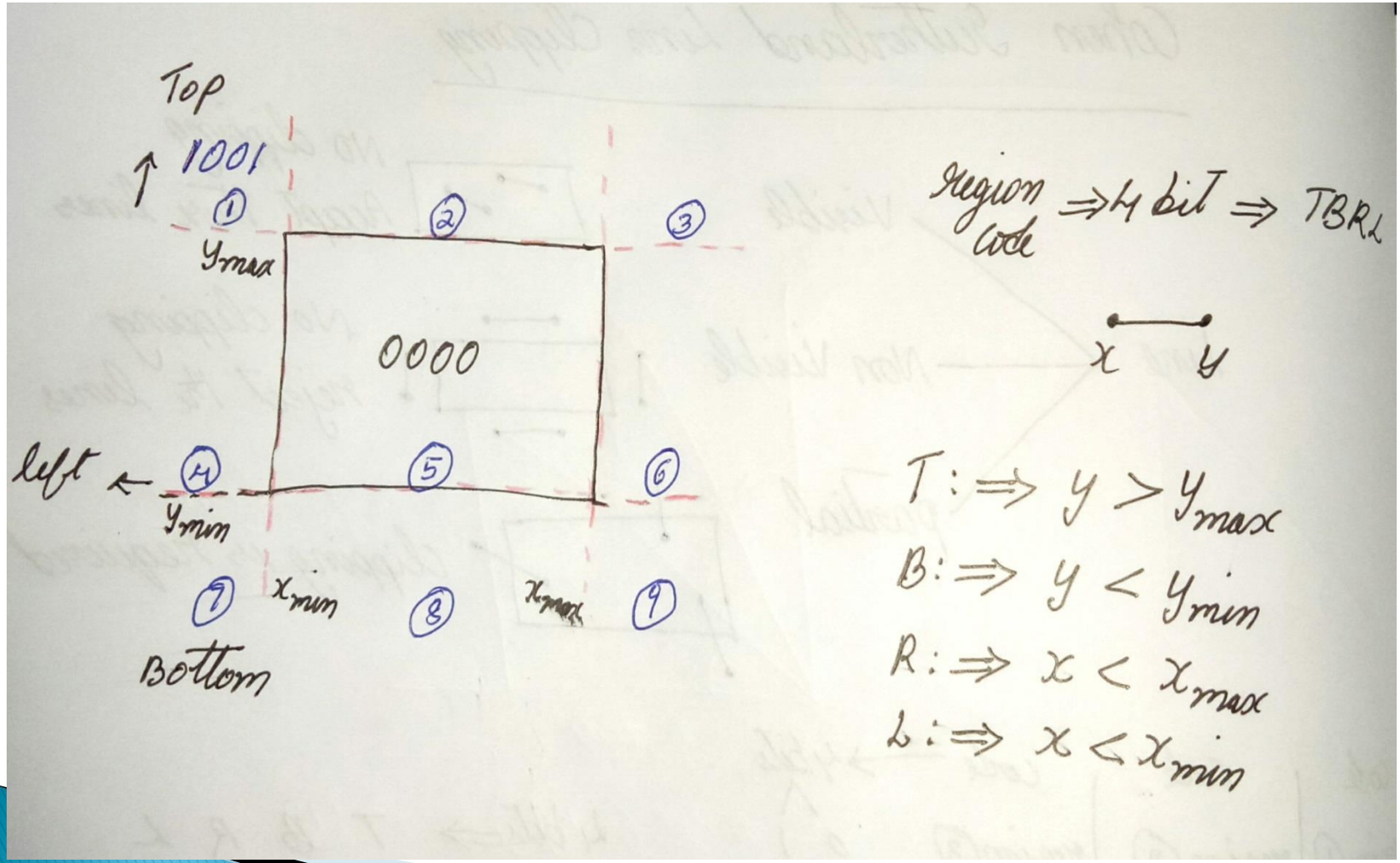
-By Manjula. S

Cohen Sutherland Line Clipping



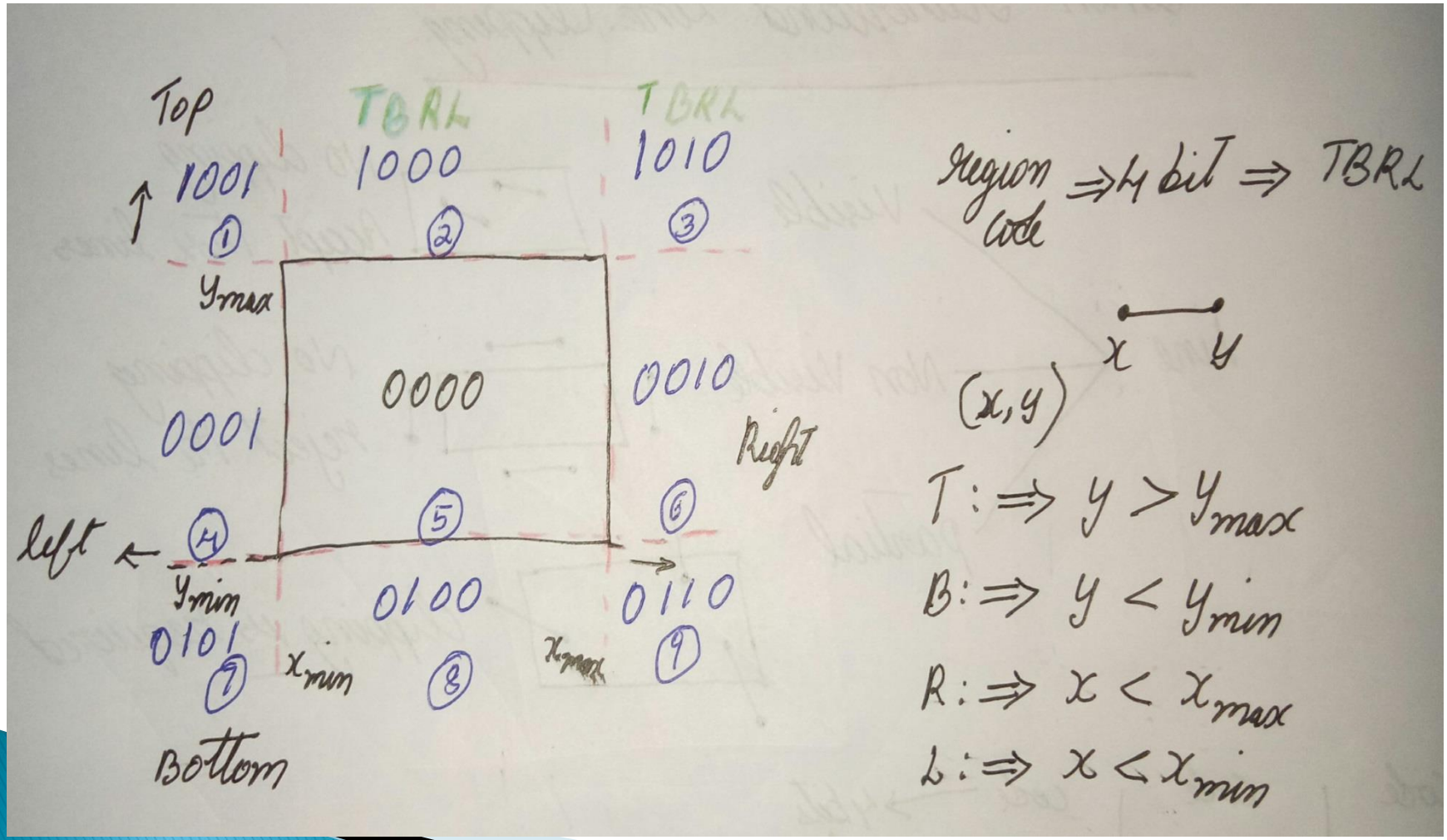
-By Manjula. S

Cohen Sutherland Line Clipping



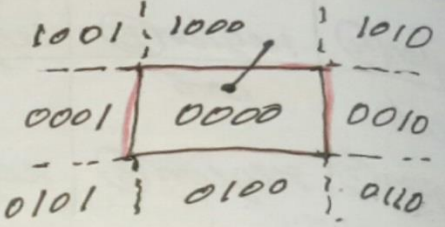
-By Manjula. S

Cohen Sutherland Line Clipping



-By Manjula. S

Cohen Sutherland Line clipping Algorithm



Step 1 : For each end points, a region code is assigned.

Step 2 : The line is accepted if both end points have a region code 0000

Step 3 : Else, the logical And operation is performed for both regions code.

Step 3.1 : If the result is not 0000 then reject the line

Step 3.2 : Else clipping is required.

Step 3.2.1 : An End point of the line is selected that is outside the window

Step 3.2.2 : Find the intersection point at the window boundary.

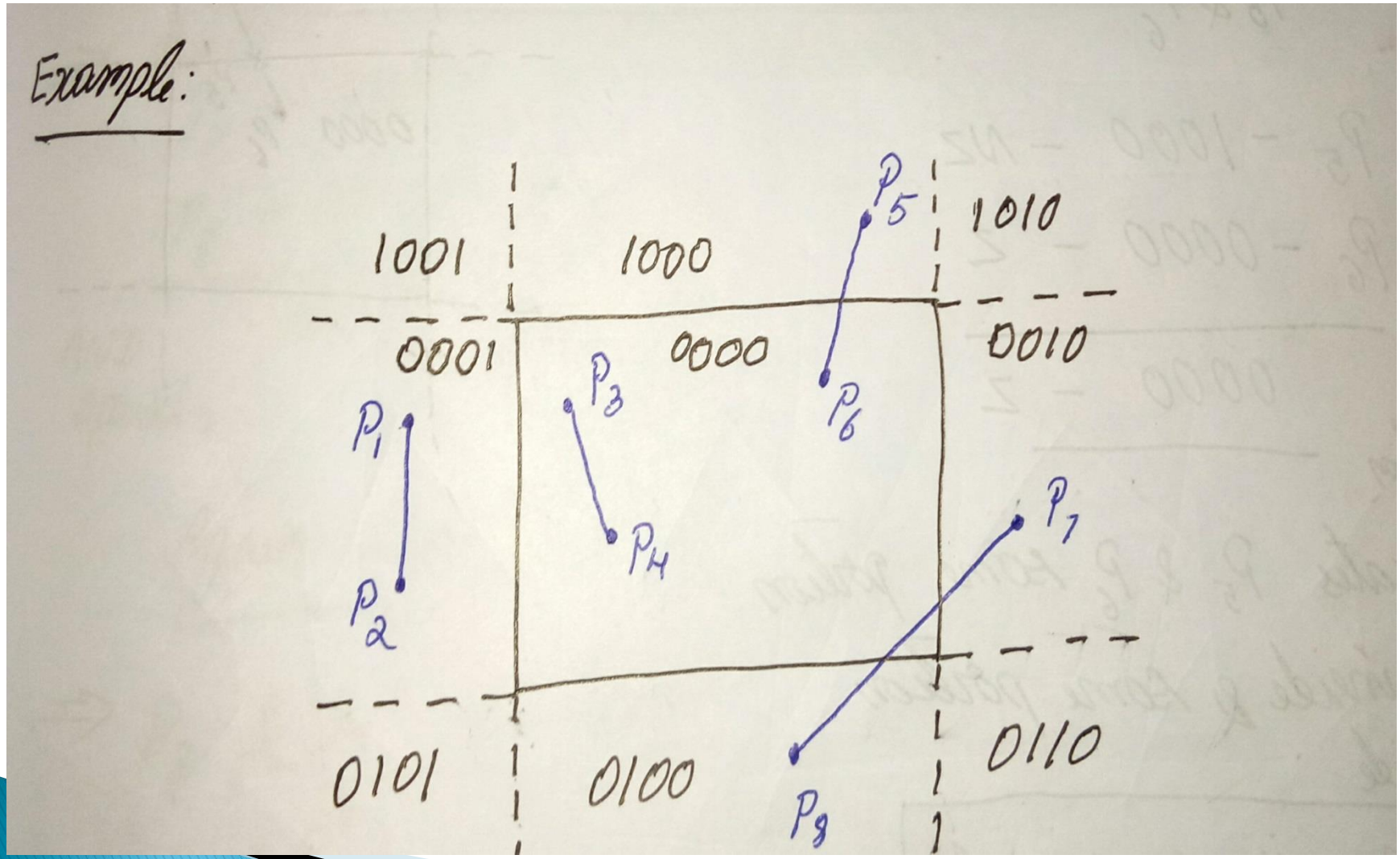
Step 3.2.3 : The end point is replaced with the intersection point & the region code is updated.

Step 3.2.4 : Repeat step 2 until we find a clipped line either trivally accepted or trivally rejected.

Step 4 : Repeat step 1 for other lines.

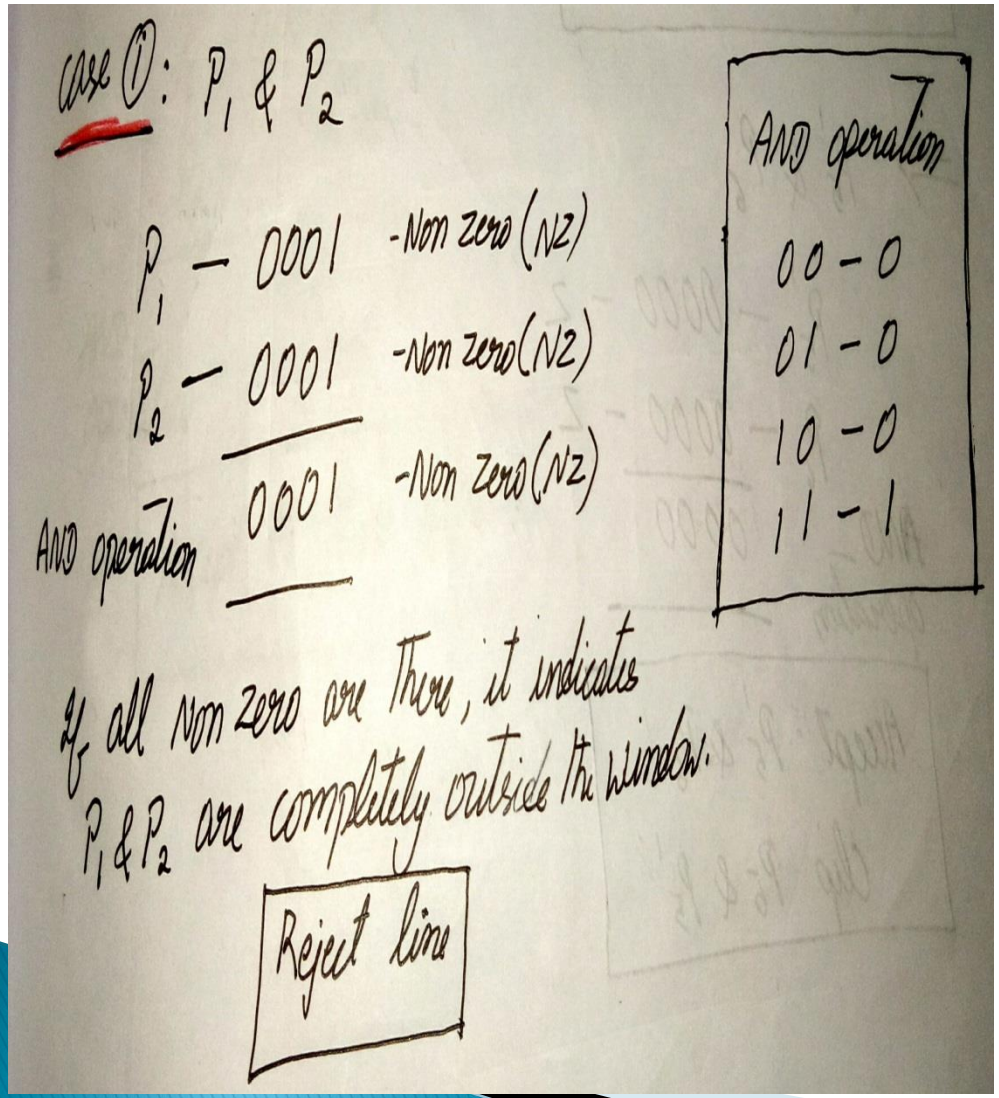
-By Manjula. S

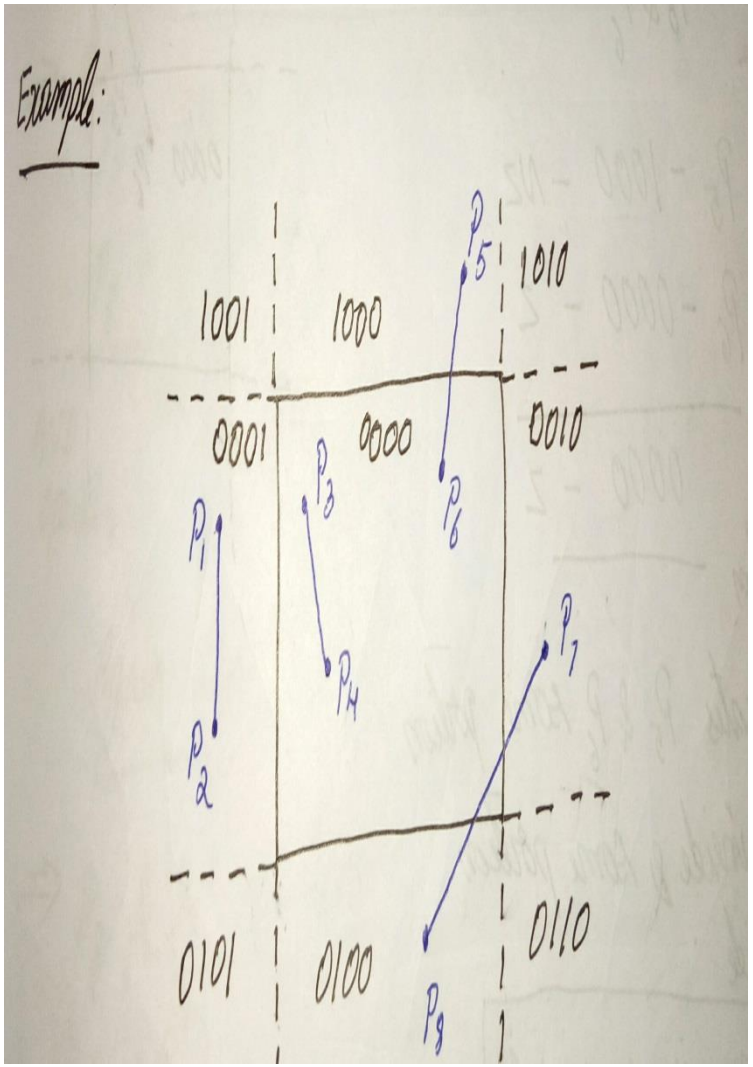
Cohen Sutherland Line Clipping



-By Manjula. S

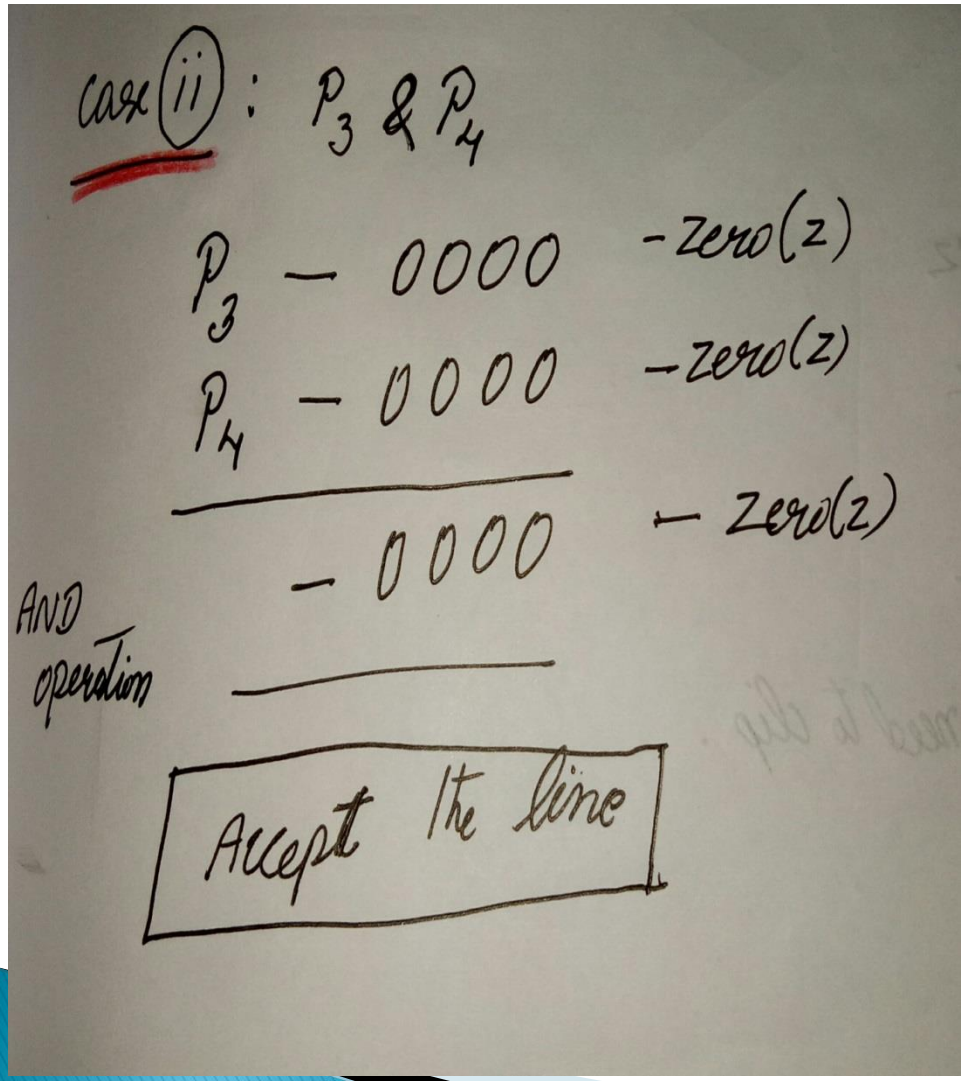
Cohen Sutherland Line Clipping

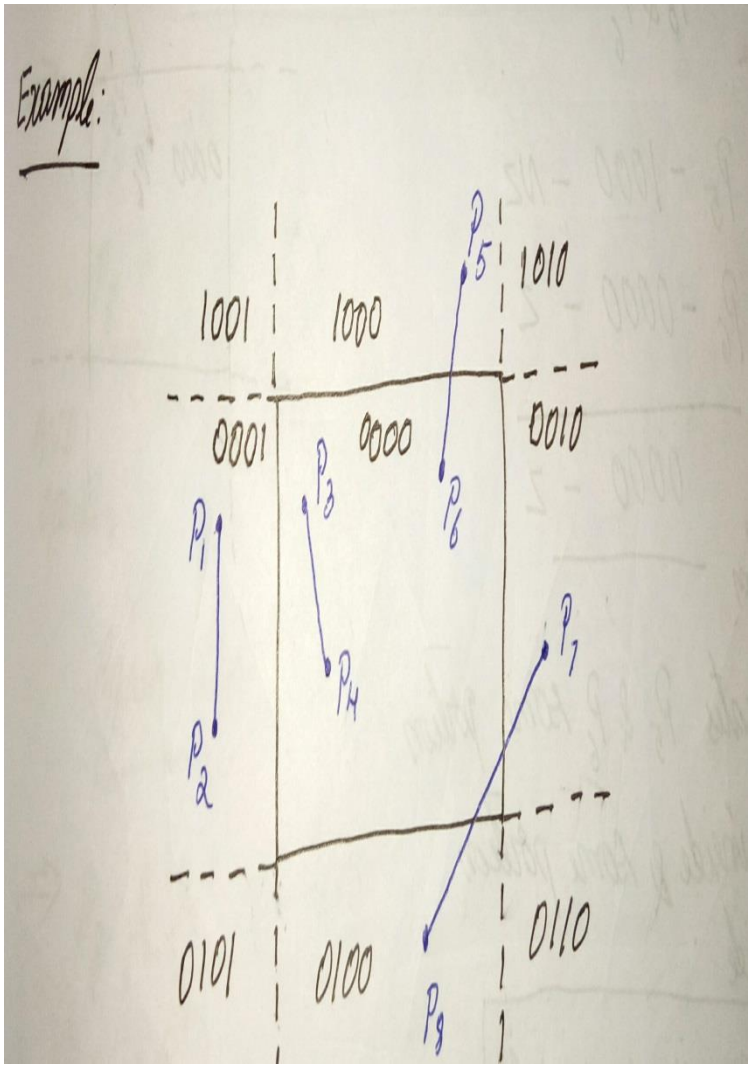




-By Manjula. S

Cohen Sutherland Line Clipping

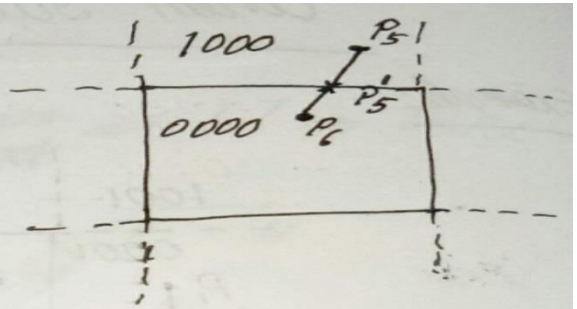




-By Manjula. S

Cohen Sutherland Line Clipping

Case (iii) : P_5 & P_6



P_5 − 1000 − NZ

P_6 − 0000 − Z

AND operation 0000 − Z

It indicates P_5 & P_6 some portion lies inside & some portion outside.

Clipping is required

$\Rightarrow$ P_5' & P_6

P_5' − 0000 − Z

P_6 − 0000 − Z

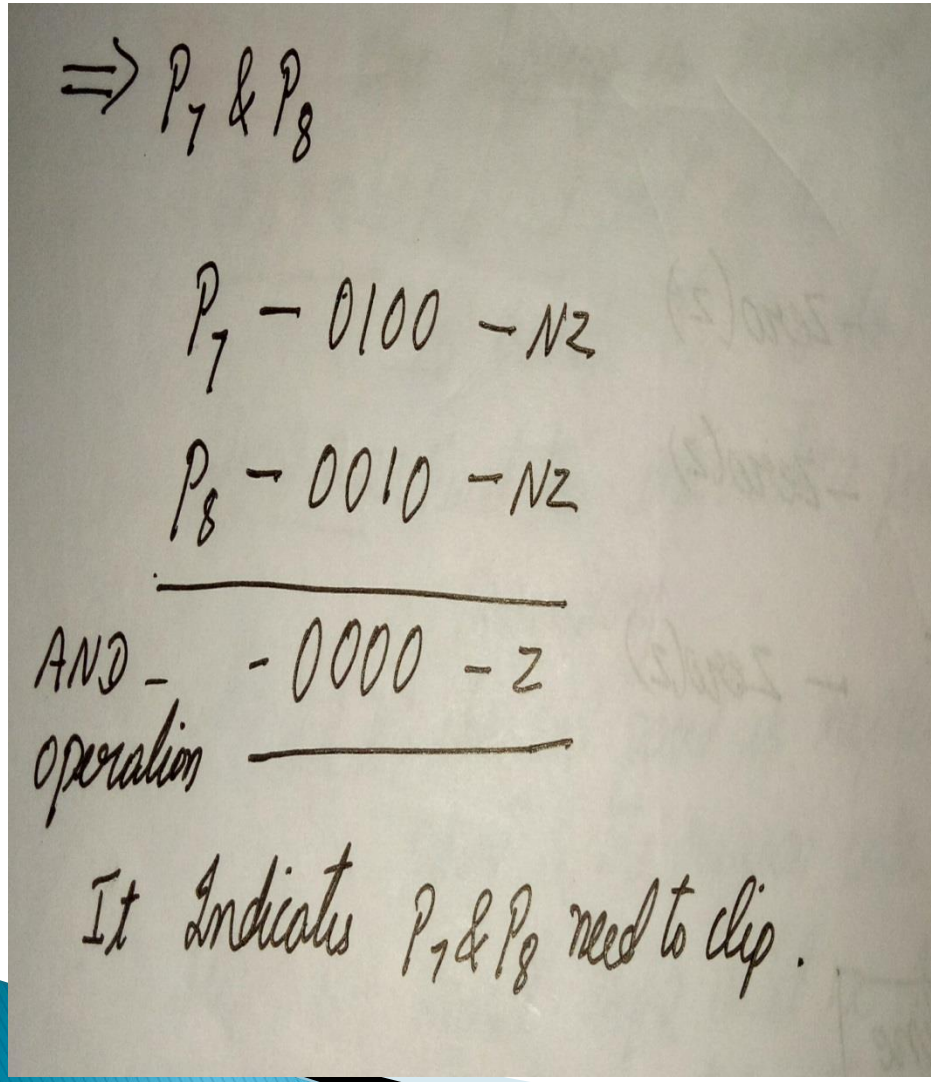
AND operation 0000

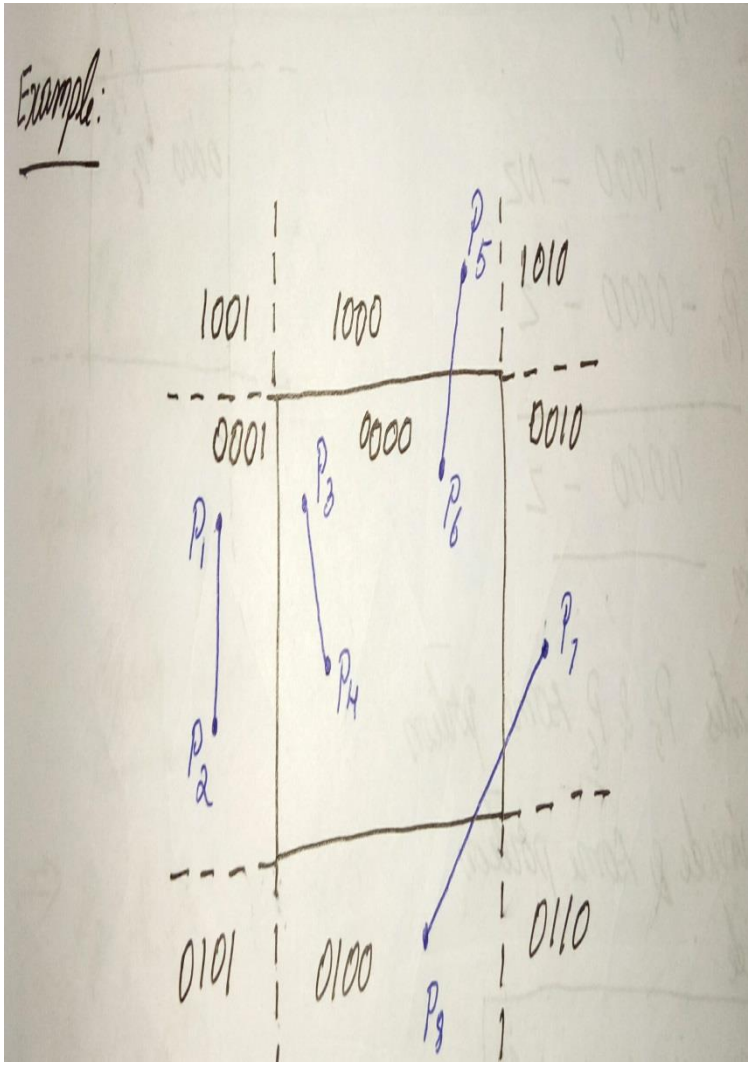
Accept P_5' & P_6

Clip P_5 & P_5'

-By Manjula. S

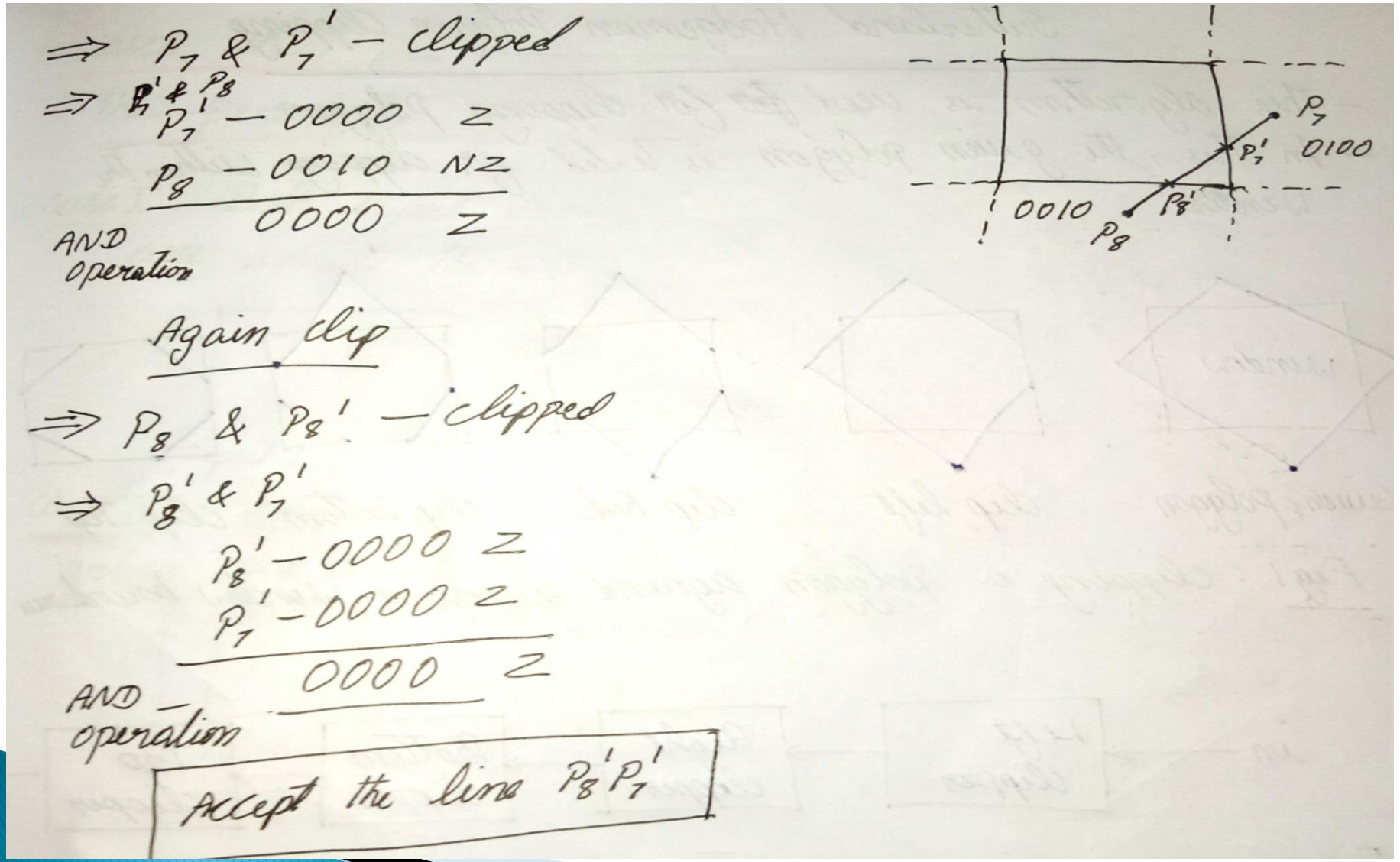
Cohen Sutherland Line Clipping





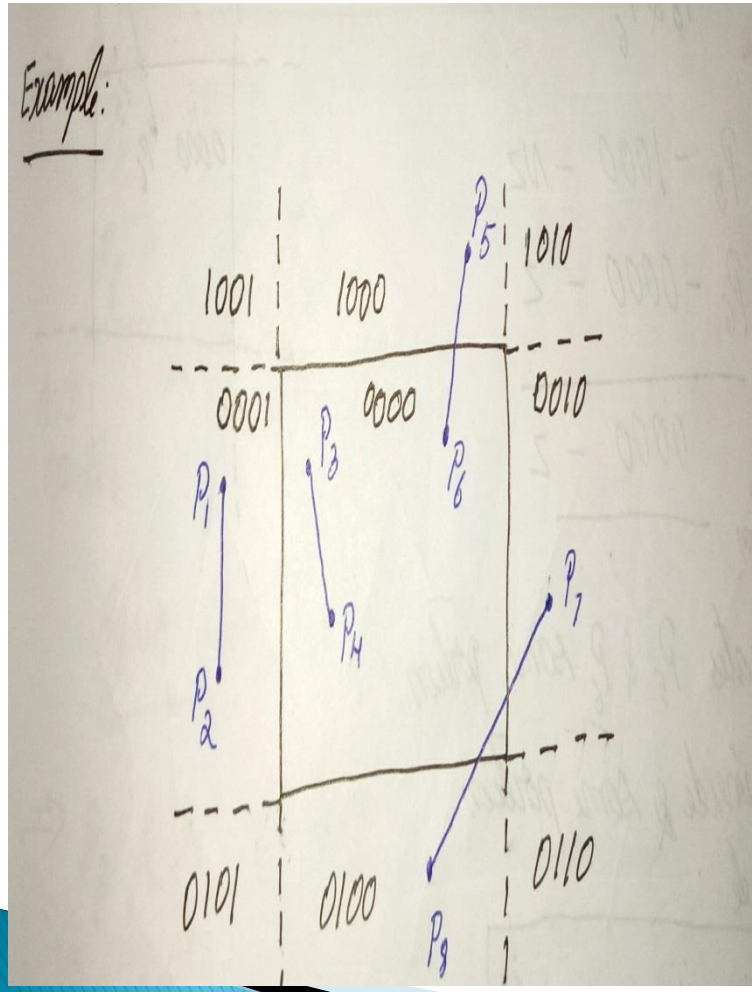
-By Manjula. S

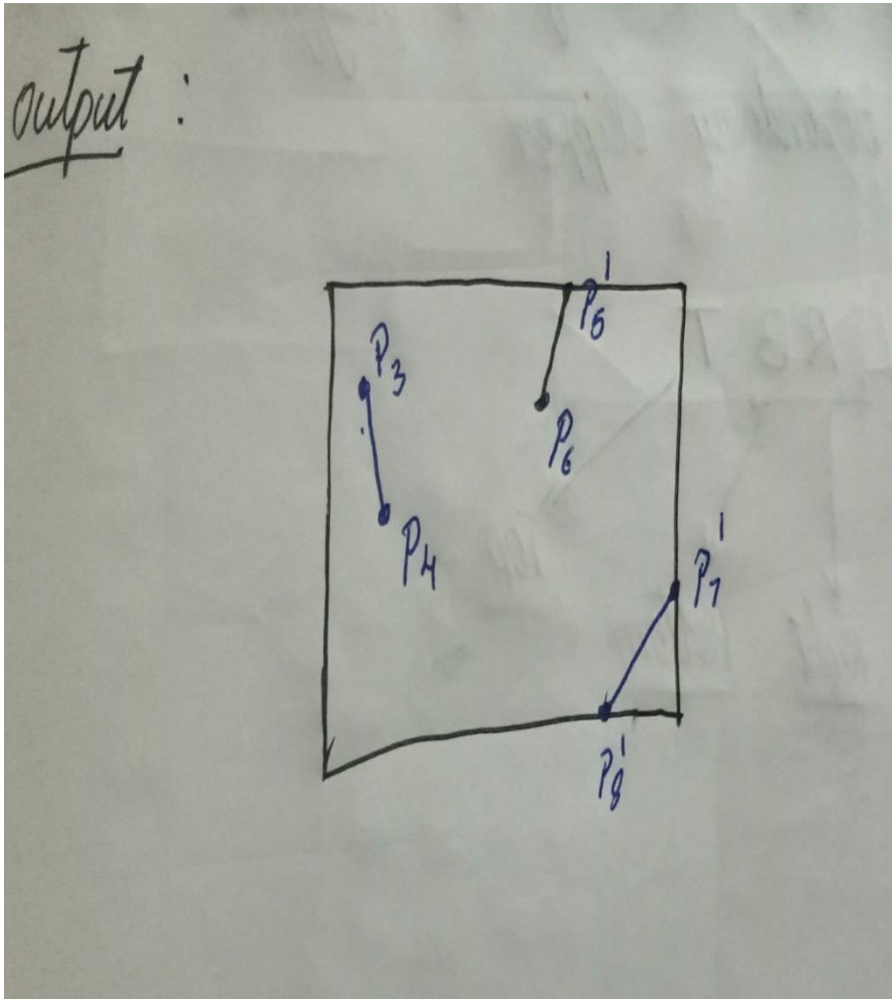
Cohen Sutherland Line Clipping



-By Manjula. S

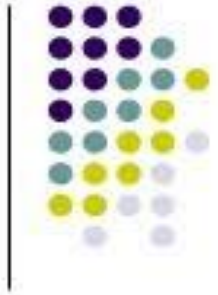
Cohen Sutherland Line Clipping





-By Manjula. S

Polygon Clipping



- Not as simple as line segment clipping
  - Clipping a line segment yields at most one line segment
  - Clipping a polygon can yield multiple polygons

- However, clipping a convex polygon can yield at most one other polygon

Sutherland-Hodgman Clipping
Rules to generate the Vertex list

- Apply the following rules for every pair of vertices
  - If the first vertex is outside and the second is inside, then move the first vertex to the clipping boundary and accept both vertices.
    - Calculate intersection point with the boundary
    - Replace the first vertex with the intersection point
    - Put the new first vertex and the second vertex in the output list
  - If both vertices are inside, put the second vertex in the output list.
  - If the first vertex is inside and the second is outside, then move the second to the clipping boundary and accept the second.
    - Calculate intersection point with the boundary
    - Replace the second vertex with the intersection point
    - Put the new second vertex in the output list
  - If both vertices are outside, reject both.

Sutherland-Hodgman Clipping Algorithm

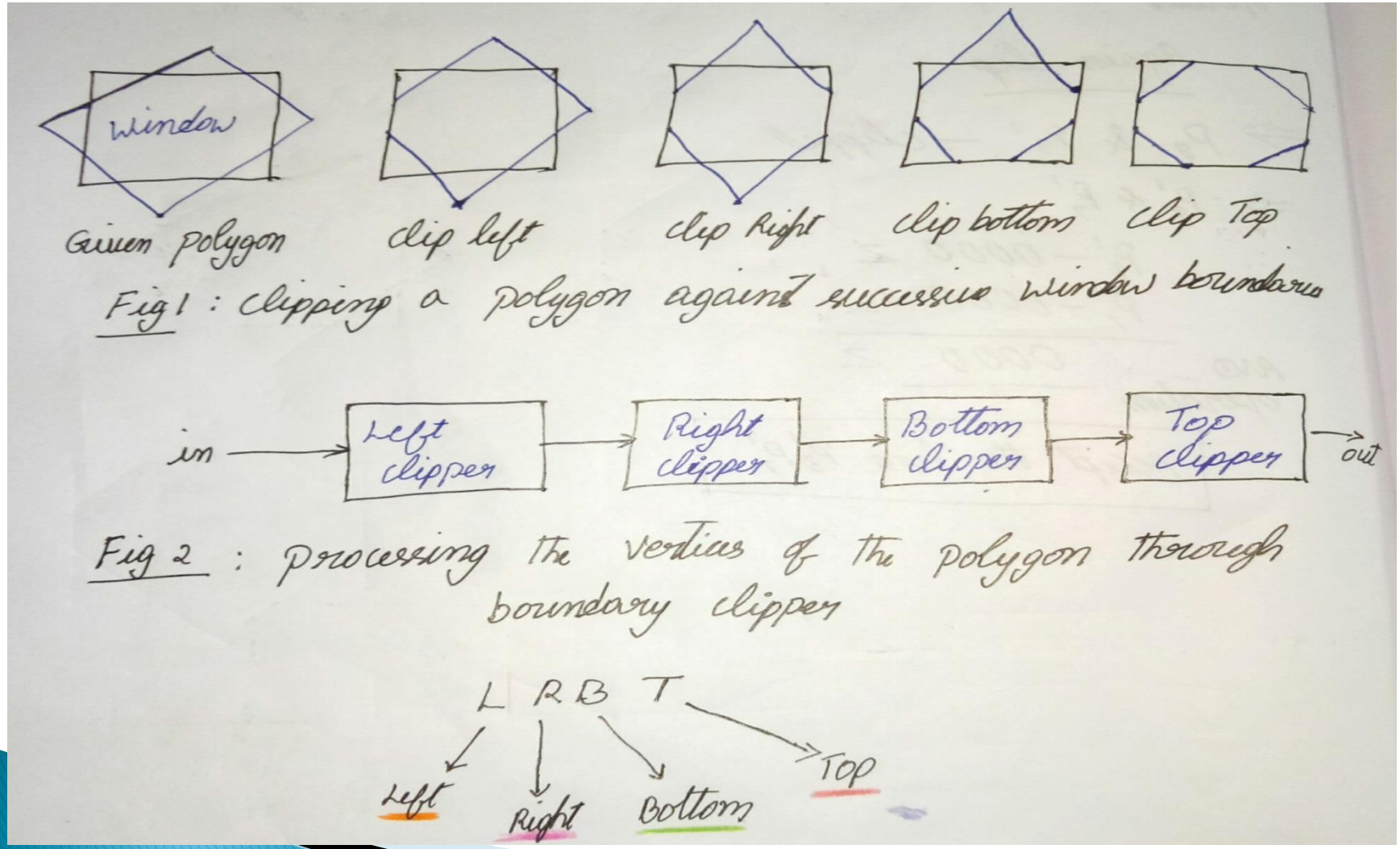


Fig 1 : clipping a polygon against successive window boundaries

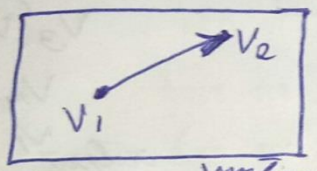
Fig 2 : processing the vertices of the polygon through boundary clipper

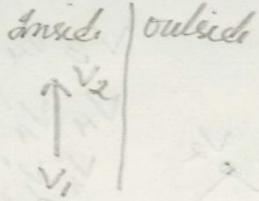
-By Manjula. S

Sutherland-Hodgman Clipping Algorithm

4 cases when processing vertices in sequence around the perimeter of polygon

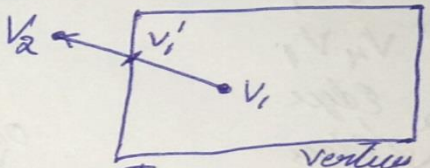
case I – Both input vertices are inside the WB

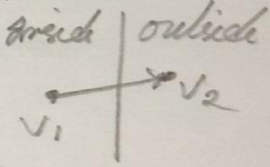




output list – V_2

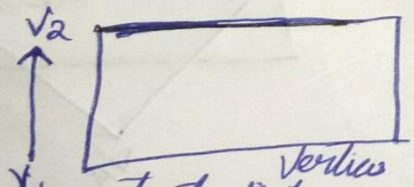
case II – First vertex is inside the WB & 2nd is outside

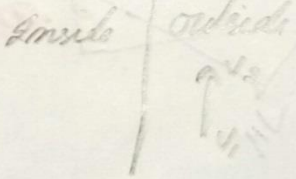




output list – V_1'

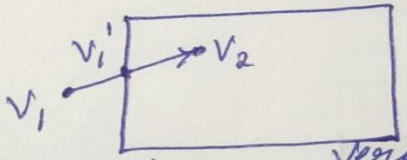
case III – If both input vertices are outside the WB

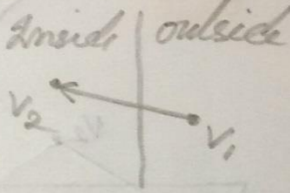




output list – Nil

case IV – First vertex is outside the WB & second vertex is inside the window.

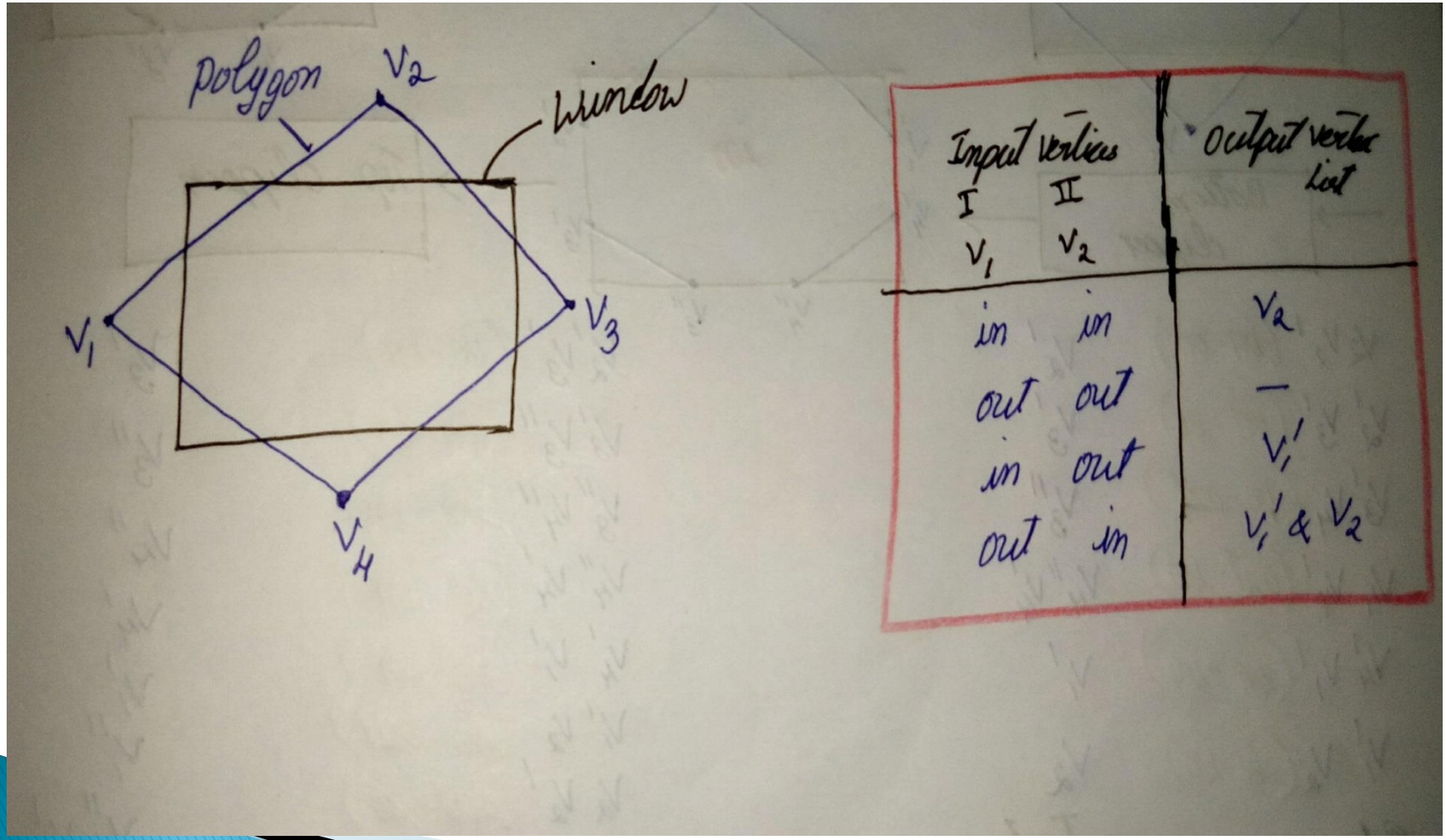




output list – V_1' & V_2

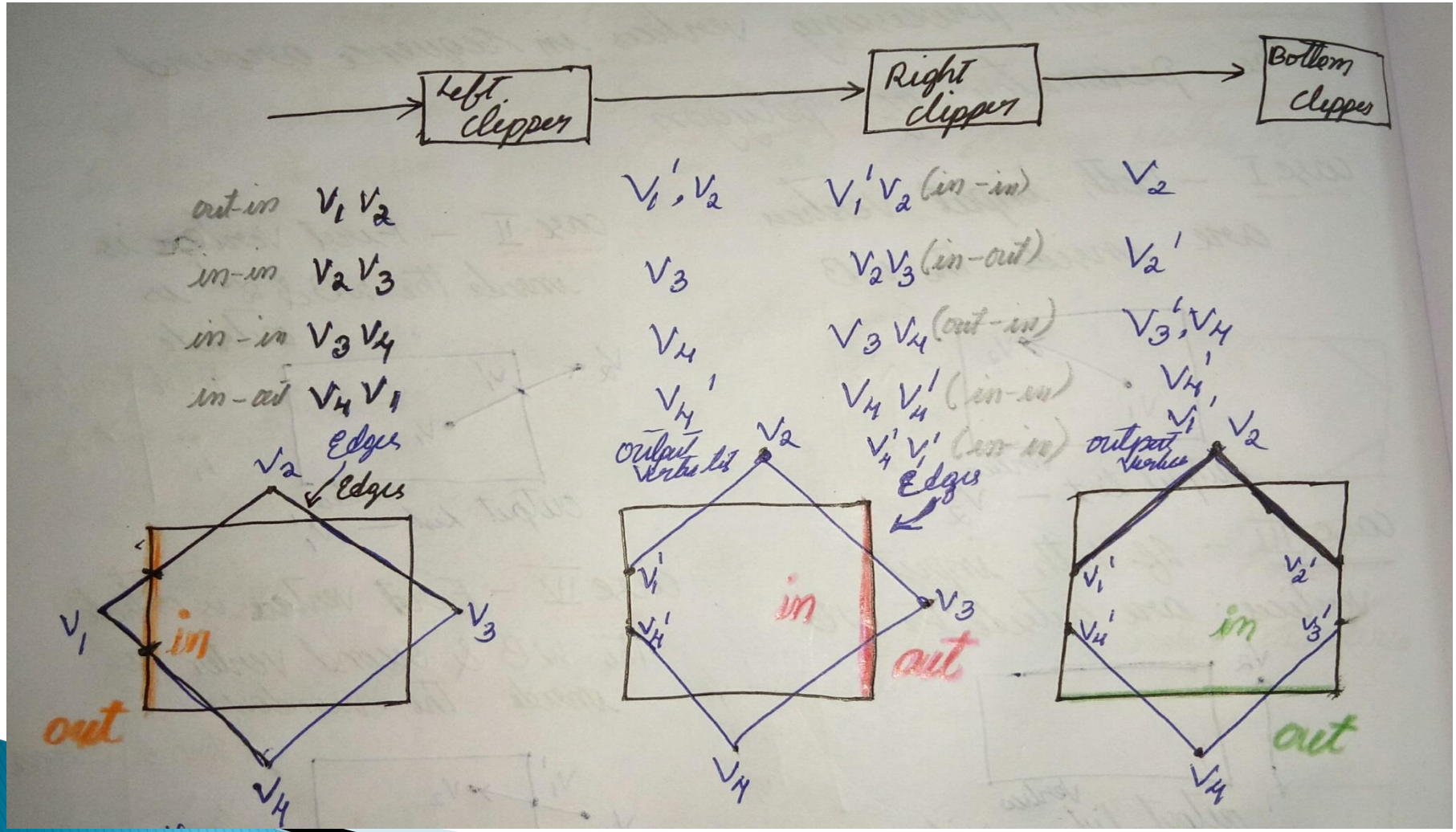
-By Manjula. S

Sutherland-Hodgman Clipping Algorithm



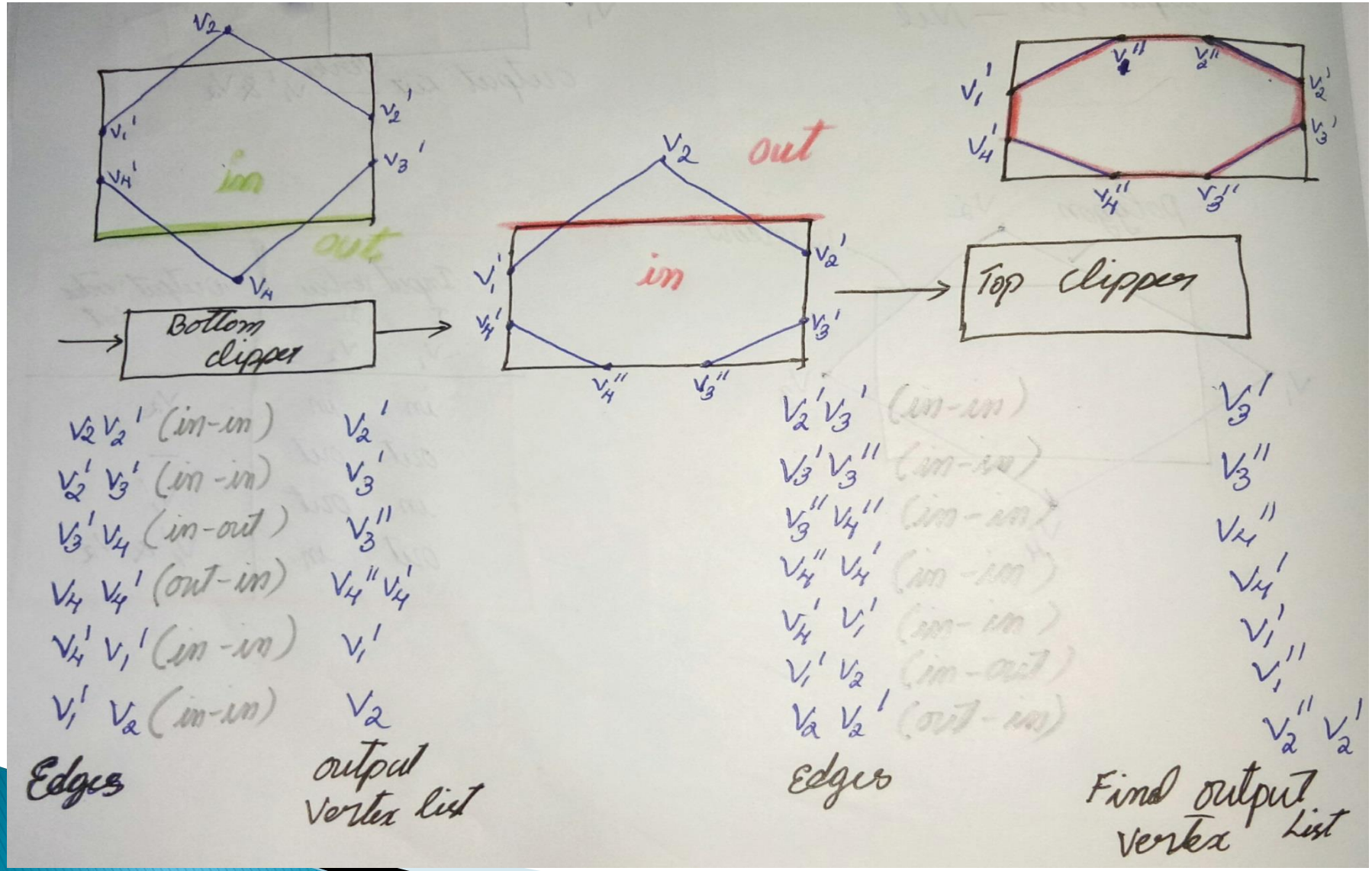
-By Manjula. S

Sutherland-Hodgman Clipping Algorithm



-By Manjula. S

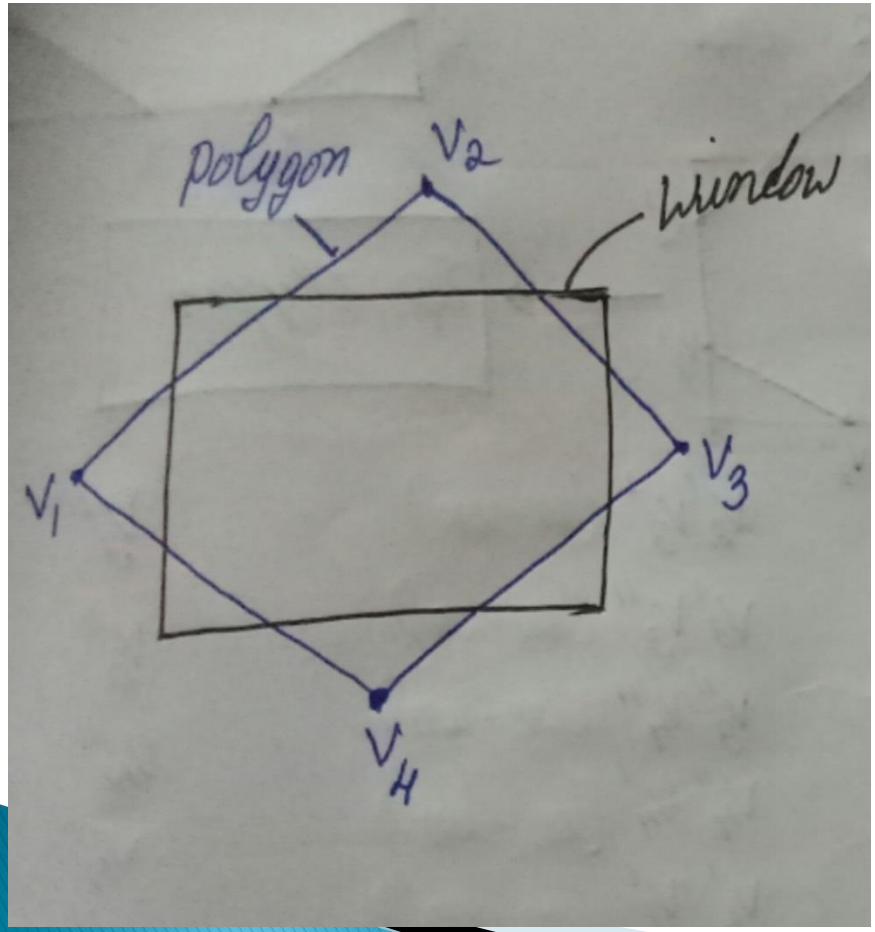
Sutherland-Hodgman Clipping Algorithm



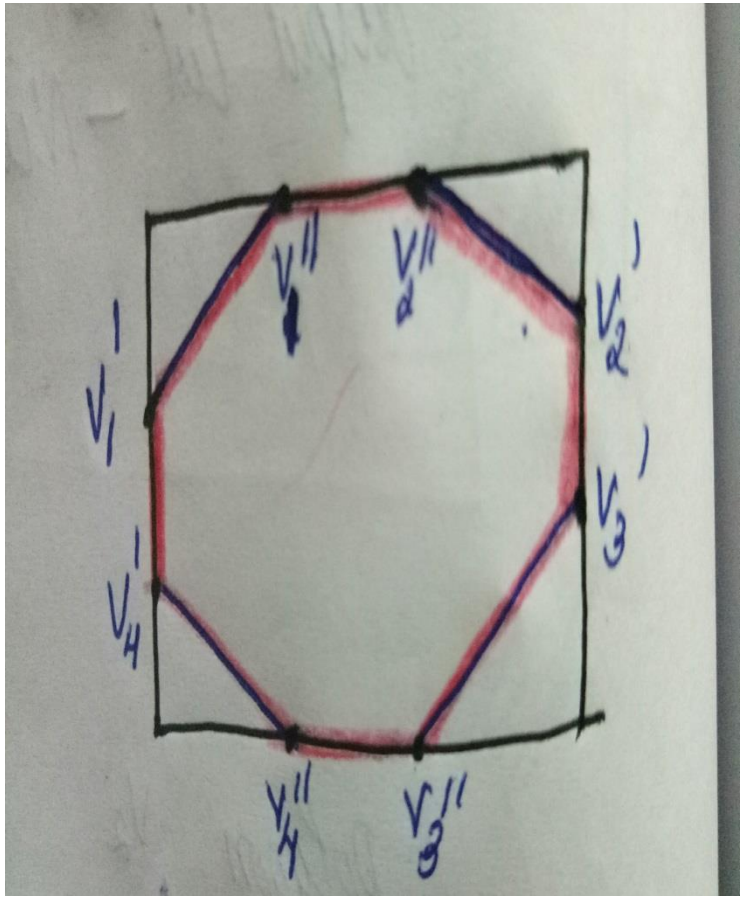
-By Manjula. S

Sutherland-Hodgman Clipping Algorithm

Input :



Output:



-By Manjula. S

Thank You

-By Manjula. S