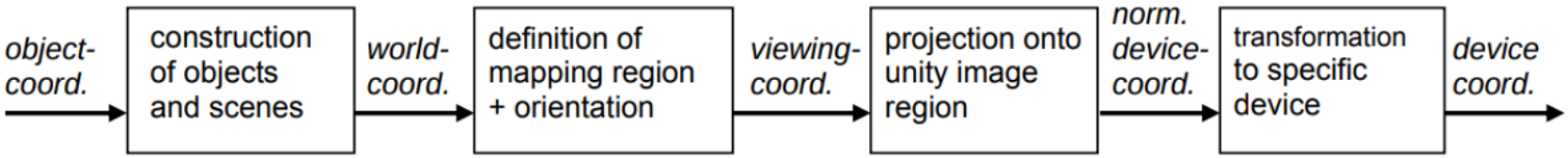
CS 461 - Computer Graphics

Viewing & Clipping

Amal Dev Parakkat

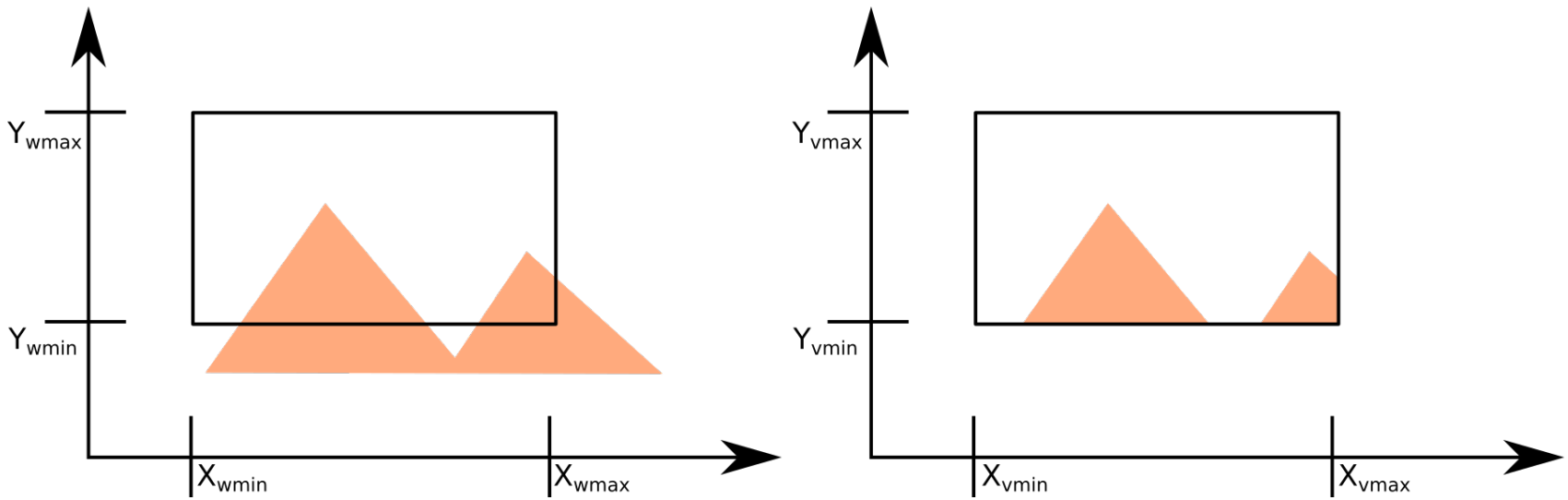
2D Viewing Pipeline



- The coordinates in which individual objects (models) are created are called model (or object) coordinates.
- When several objects are assembled into a scene, they are described by world coordinates.
- After transformation into the coordinate system of the camera (viewer) they become viewing coordinates.
- Their projection onto a common plane (window) yields device-independent normalized coordinates.
- After mapping those normalized coordinates to a specific device, we get device coordinates.

Window to Viewport

- Window - A world-coordinate area selected for display
- Clipping window - the part of two dimensional scene to be displayed
- Viewport - window where data from clipping window will be displayed



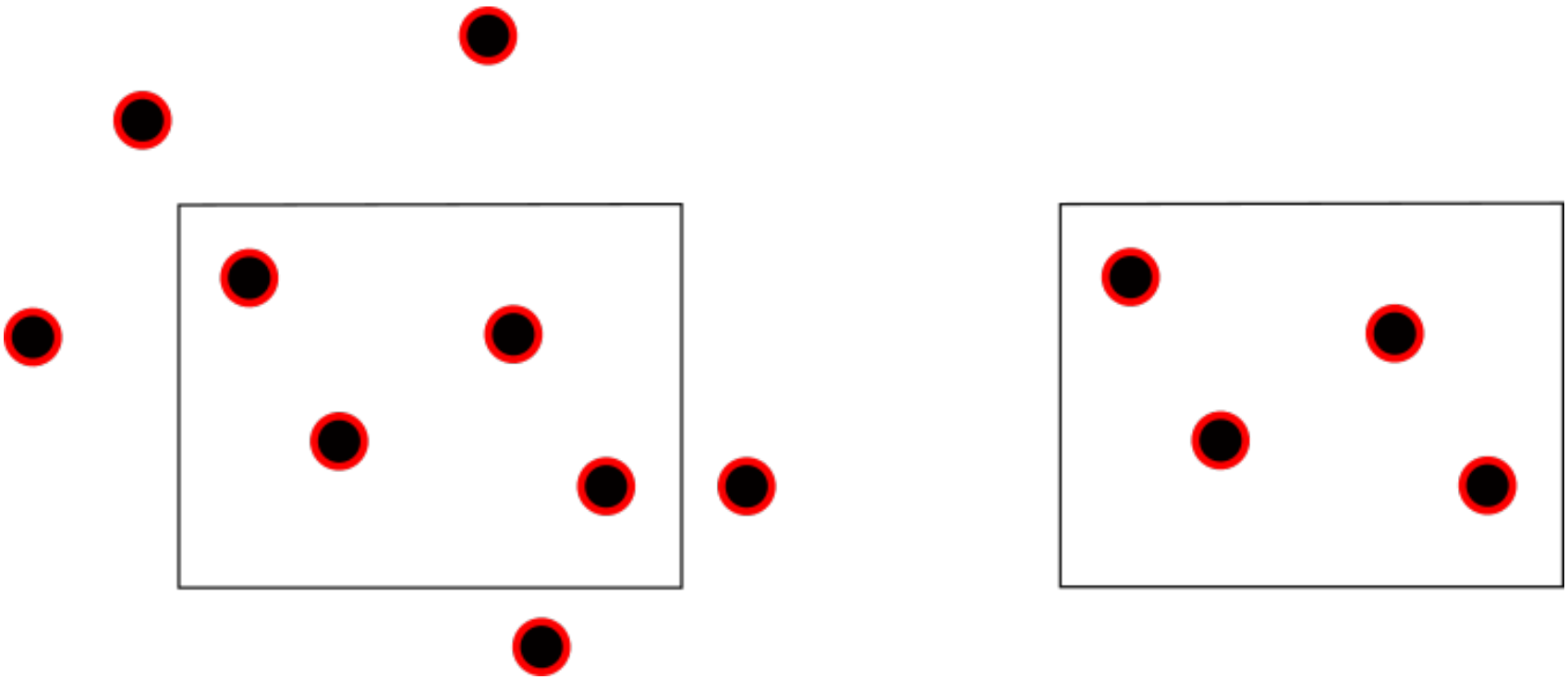
Window to Viewport mapping

- $(x_w, y_w) \longrightarrow (x_v, y_v)$
- Normalized point on window = $(\frac{x_w - x_{wmin}}{x_{wmax} - x_{wmin}}, \frac{y_w - y_{wmin}}{y_{wmax} - y_{wmin}})$
- Normalized point on viewport = $(\frac{x_v - x_{vmin}}{x_{vmax} - x_{vmin}}, \frac{y_v - y_{vmin}}{y_{vmax} - y_{vmin}})$
- $\frac{x_w - x_{wmin}}{x_{wmax} - x_{wmin}} = \frac{x_v - x_{vmin}}{x_{vmax} - x_{vmin}}, \frac{y_w - y_{wmin}}{y_{wmax} - y_{wmin}} = \frac{y_v - y_{vmin}}{y_{vmax} - y_{vmin}}$
- $x_v = x_{vmin} + (x_w - x_{wmin}) * s_x, y_v = y_{vmin} + (y_w - y_{wmin}) * s_y$
- $s_x = \frac{x_{vmax} - x_{vmin}}{x_{wmax} - x_{wmin}}, s_y = \frac{y_{vmax} - y_{vmin}}{y_{wmax} - y_{wmin}}$

Clipping

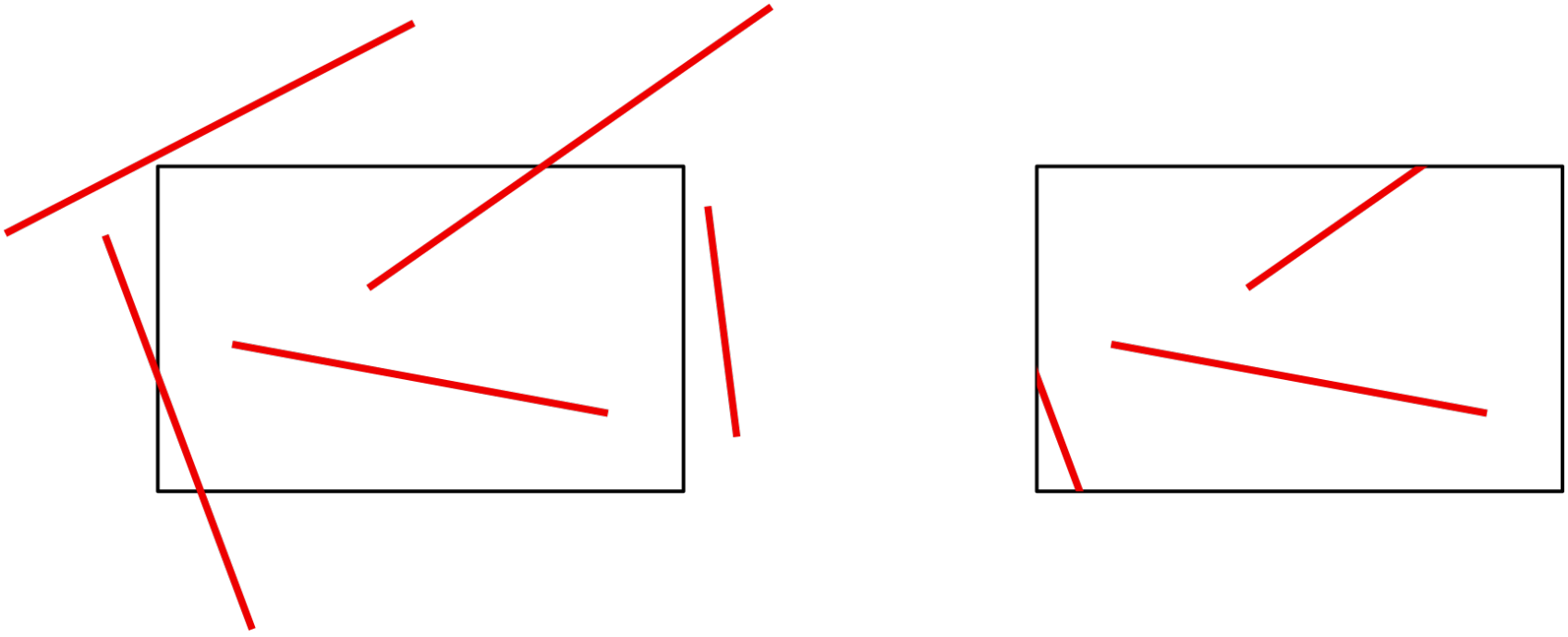
- Clipping algorithms:
  - Point clipping
  - Line clipping
  - Polygon clipping

Point clipping

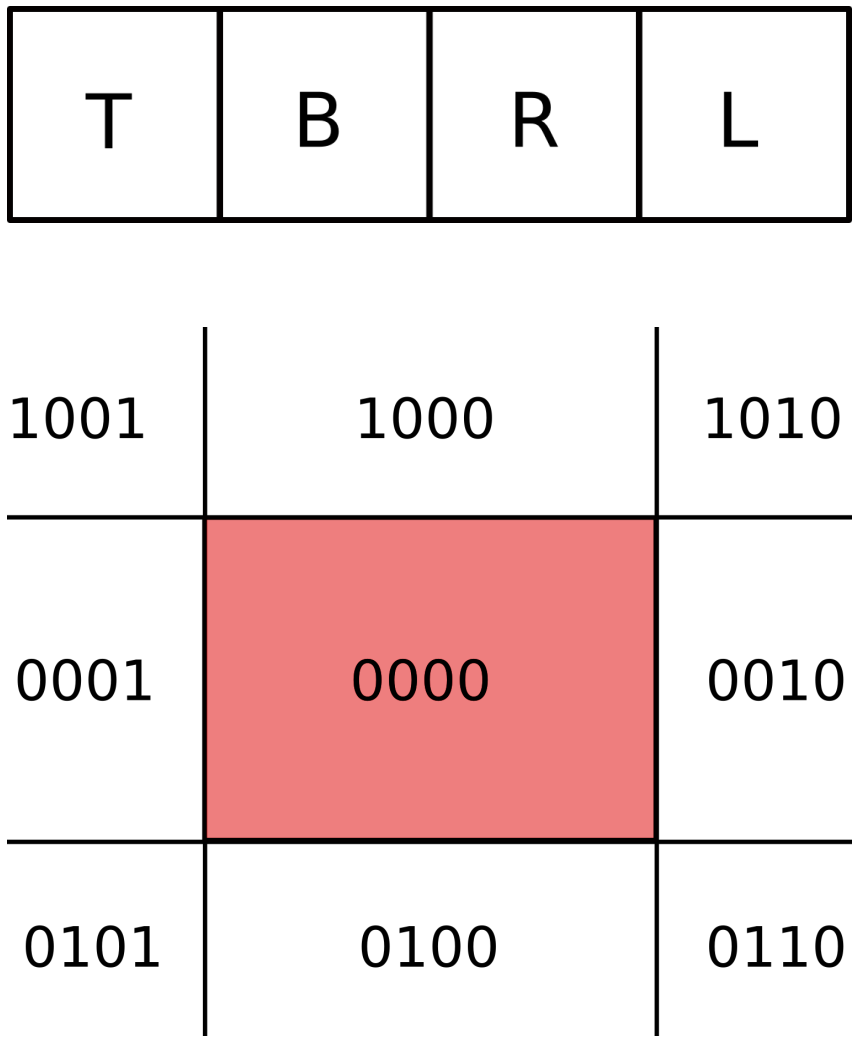


Line Clipping

- Cohen-Sutherland Line Clipping
- Nicholl-Lee-Nicholl Line Clipping

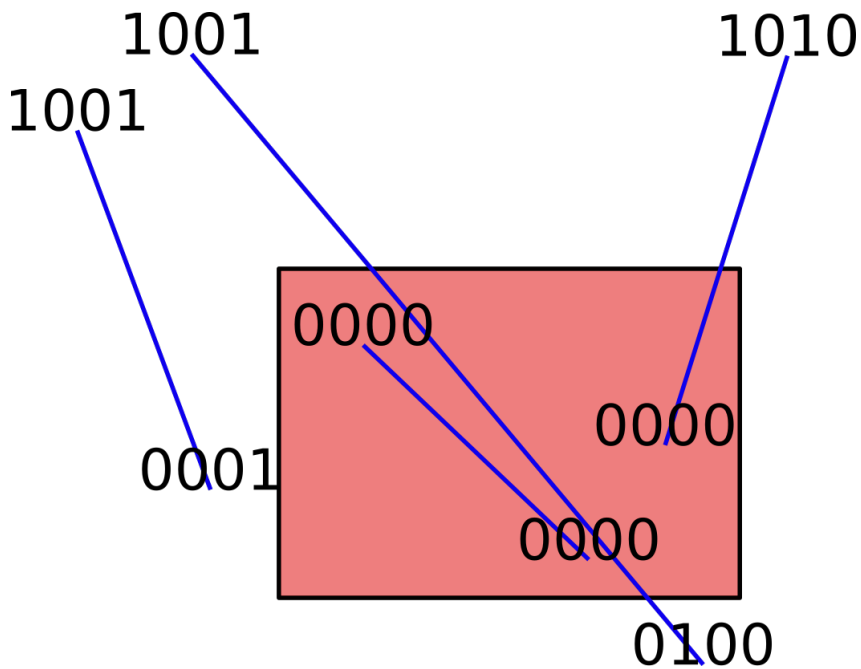


Cohen-Sutherland Line Clipping - idea



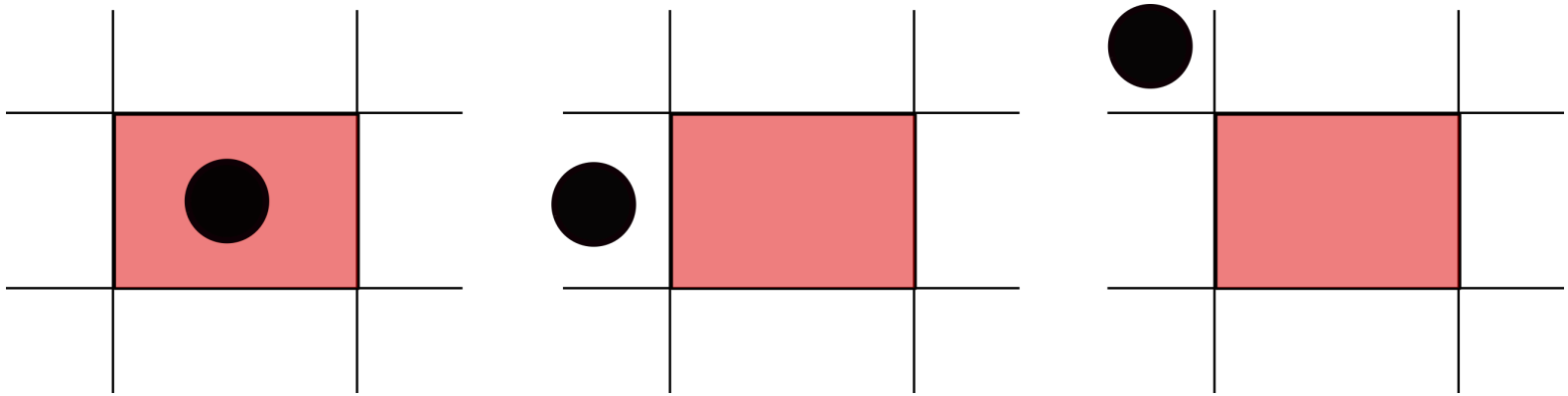
Cohen-Sutherland Line Clipping - procedure

- Find the properties
- Using OR and AND operations
- Procedure

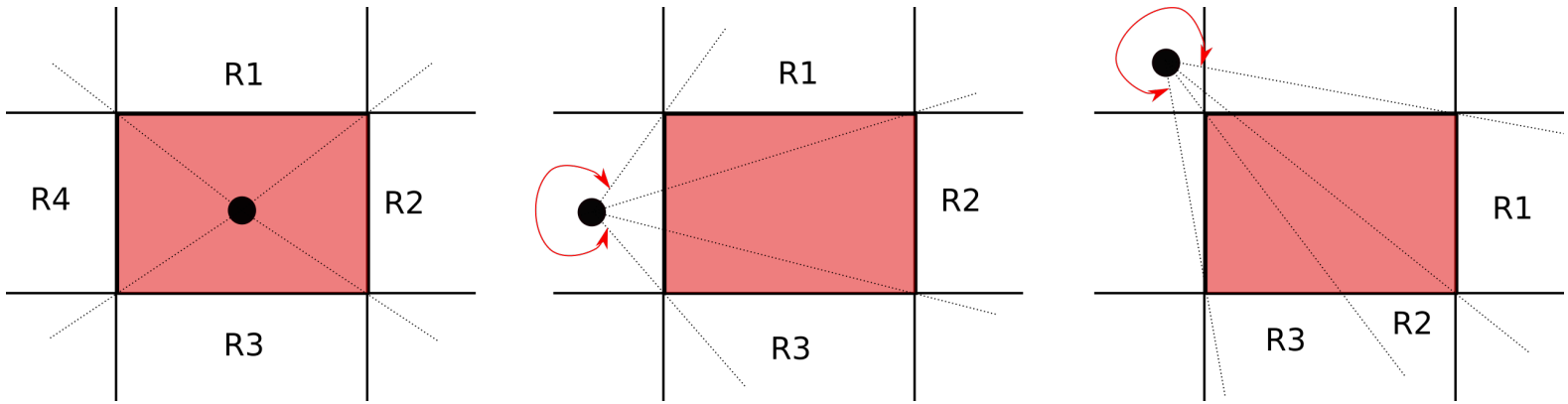


Nicholl-Lee-Nicholl Line Clipping

- Three configurations

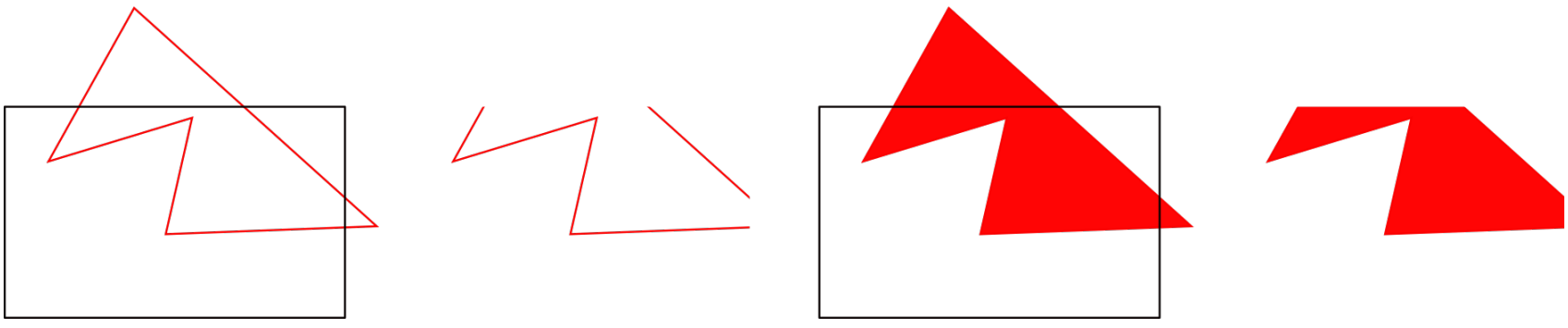


- Procedure:

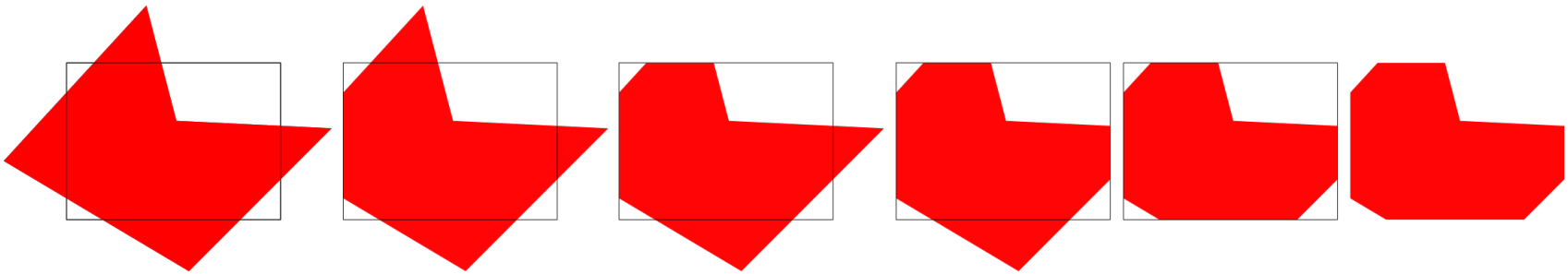


Polygon Clipping

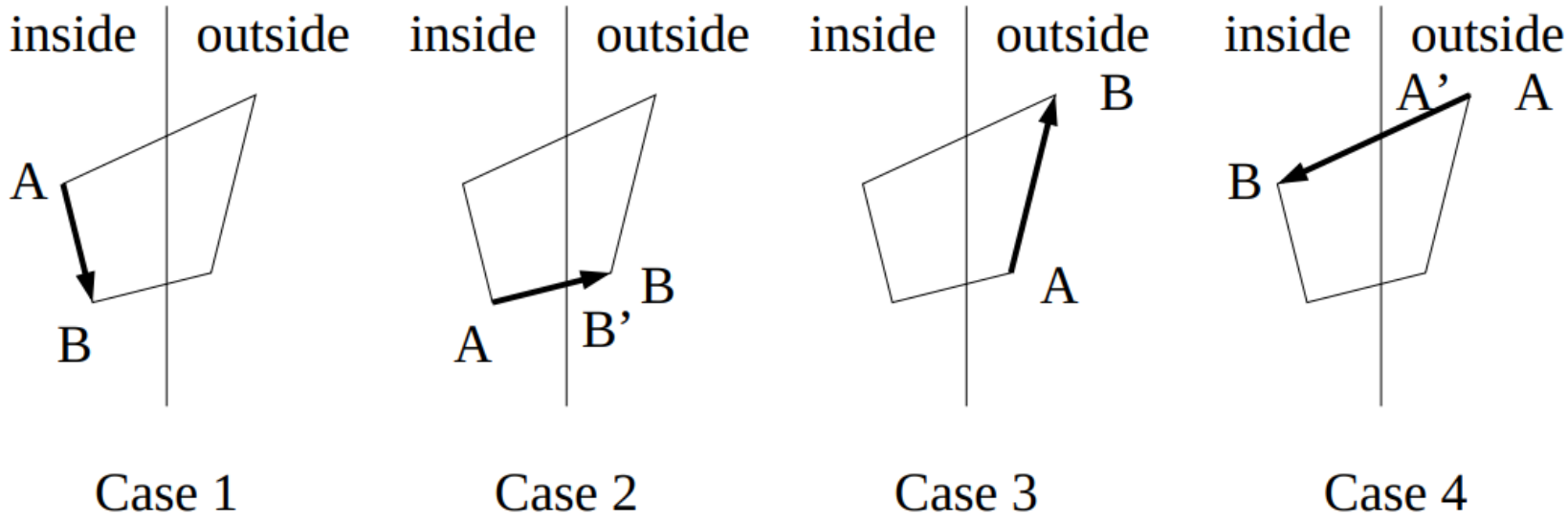
- Sutherland-Hodgman Polygon Clipping
- Weiler-Atherton Polygon Clipping



Sutherland-Hodgman Polygon Clipping - idea

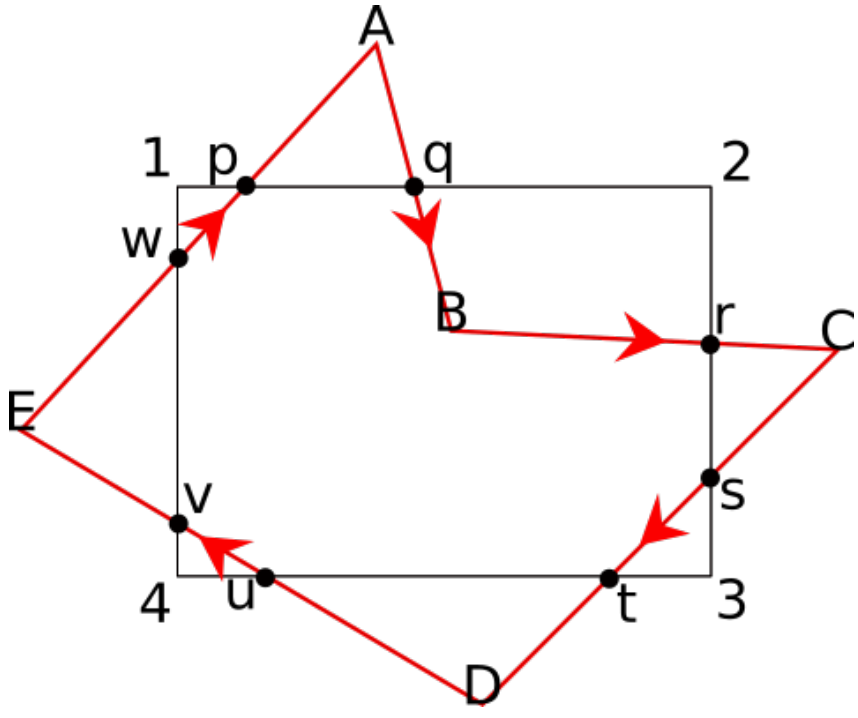


Sutherland-Hodgman Polygon Clipping - procedure



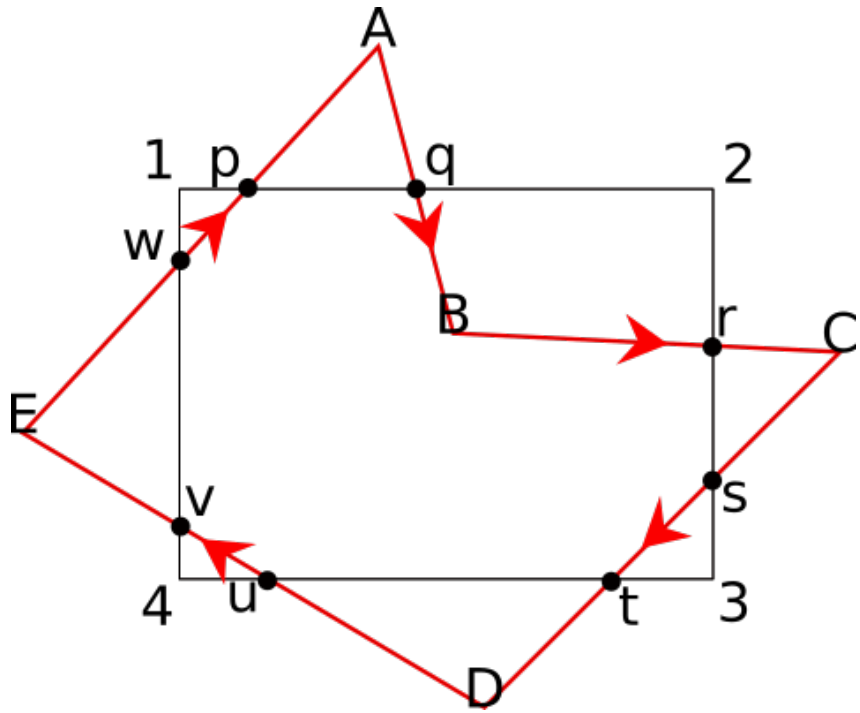
- Case 1 — vertex B is added to the output list
- Case 2 — vertex B' is added to the output (edge AB is clipped to AB')
- Case 3 — no vertex added (segment AB clipped out)
- Case 4 — vertices A' and B are added to the output (edge AB is clipped to A'B)

Sutherland-Hodgman Polygon Clipping - example



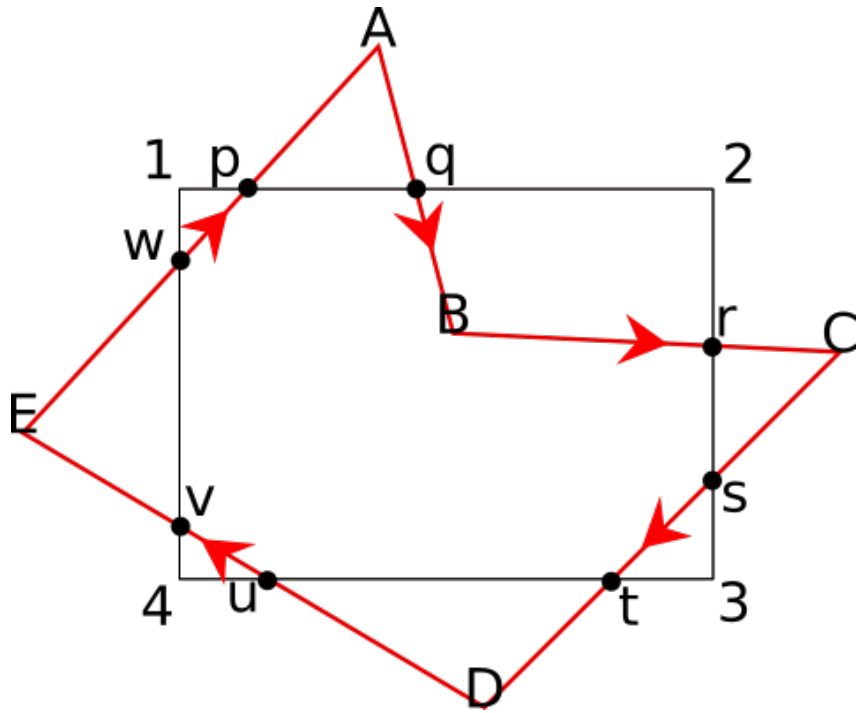
- With respect to line 1-2
- Output=ϕ

Sutherland-Hodgman Polygon Clipping - example



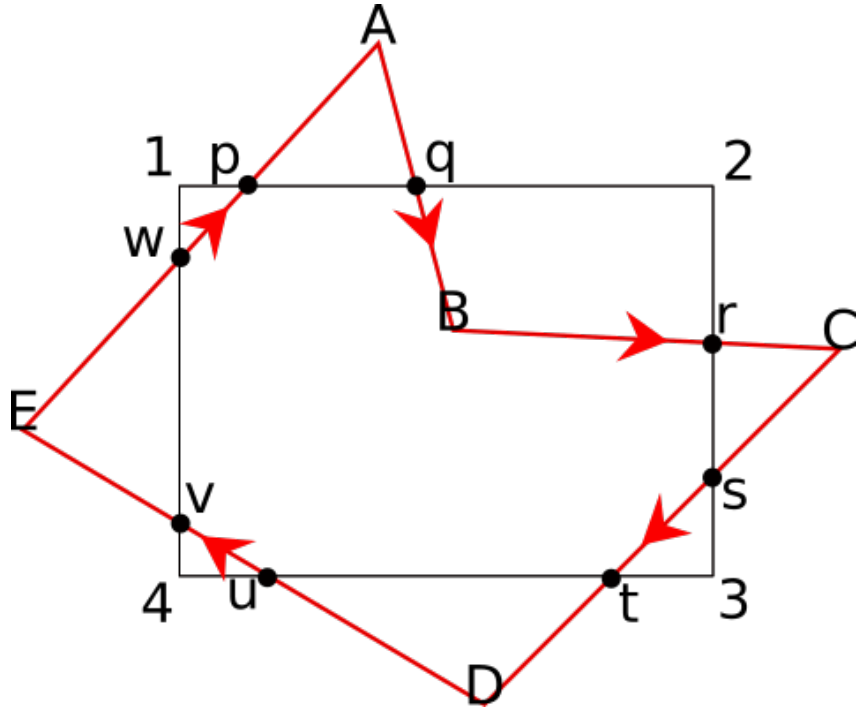
- With respect to line 1-2
- Line segment AB $\longrightarrow$ Case 4
- Output=qB

Sutherland-Hodgman Polygon Clipping - example



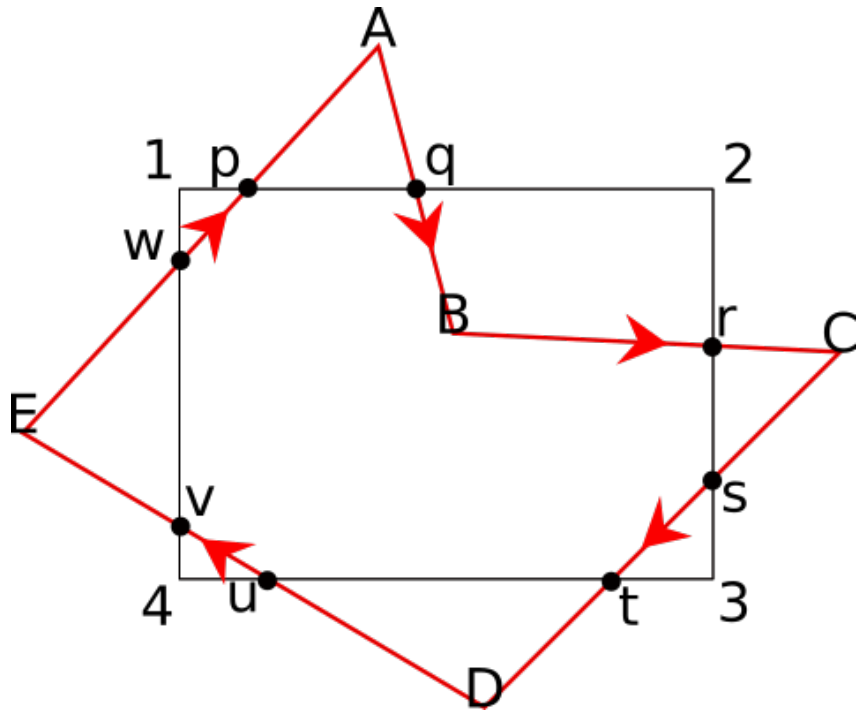
- With respect to line 1-2
- Line segment BC $\longrightarrow$ Case 1
- Output=qBC

Sutherland-Hodgman Polygon Clipping - example



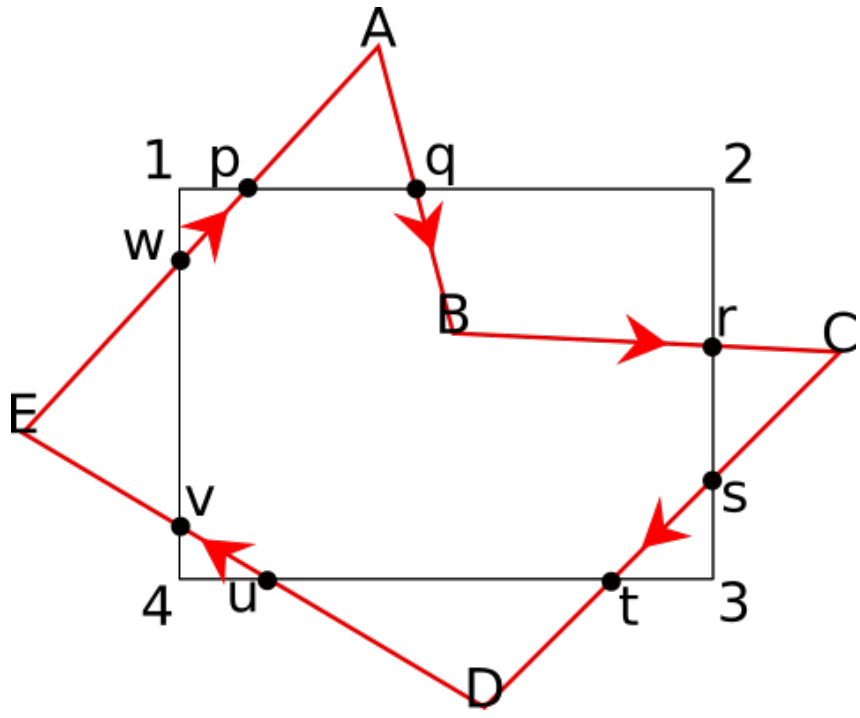
- With respect to line 1-2
- Line segment CD $\longrightarrow$ Case 1
- Output=qBCD

Sutherland-Hodgman Polygon Clipping - example



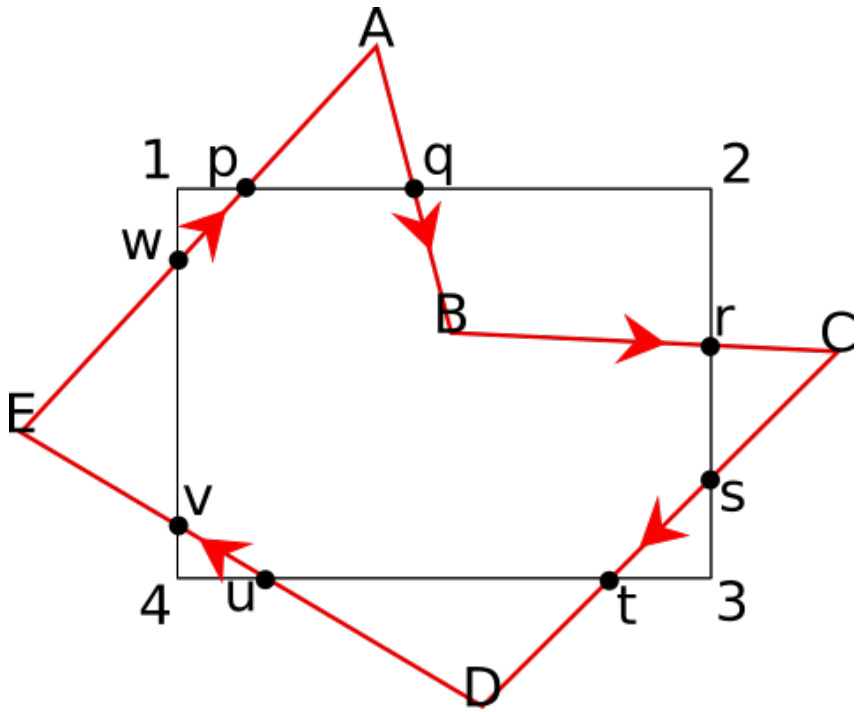
- With respect to line 1-2
- Line segment DE $\longrightarrow$ Case 1
- Output=qBCDE

Sutherland-Hodgman Polygon Clipping - example



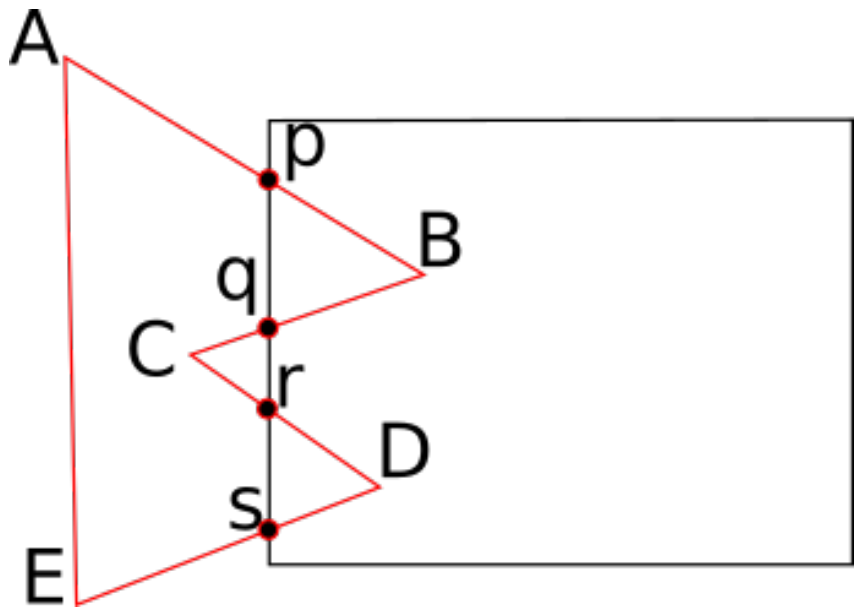
- With respect to line 1-2
- Line segment EA $\longrightarrow$ Case 2
- Output=qBCDEp

Sutherland-Hodgman Polygon Clipping - example



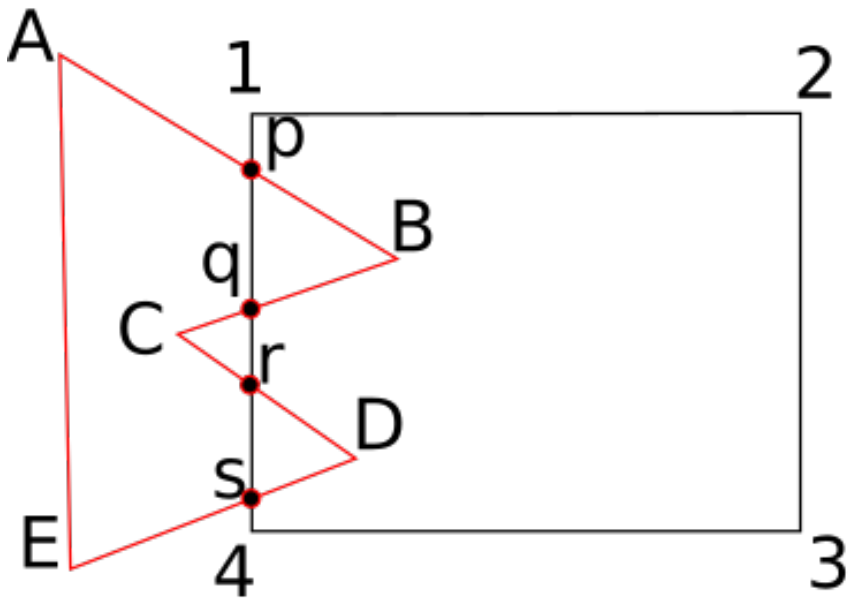
- Similarly we will do for 2-3, 3-4, and 4-1
- Output = qBrstuvwp

Sutherland-Hodgman Polygon Clipping - limitation



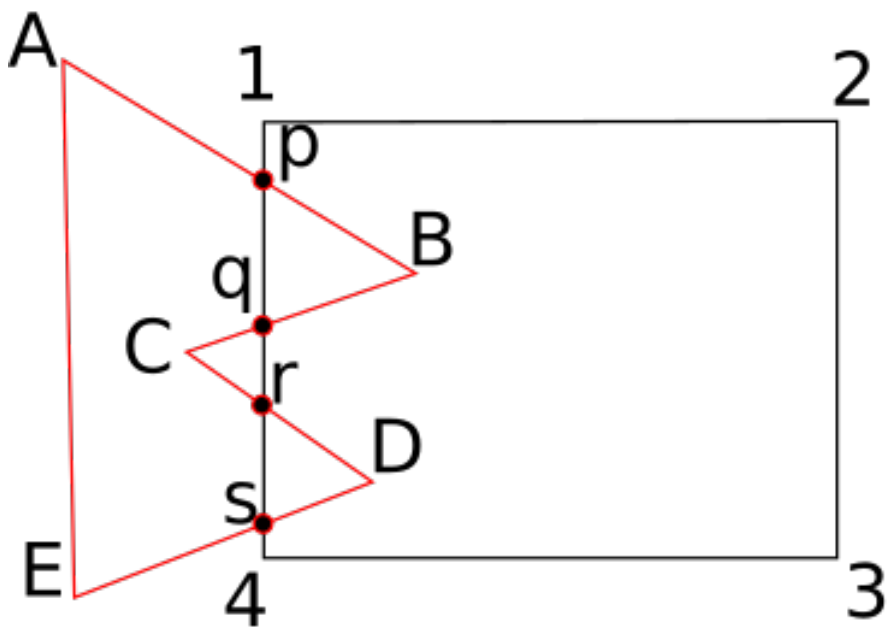
- Concave polygons
- Concave clipping polygon

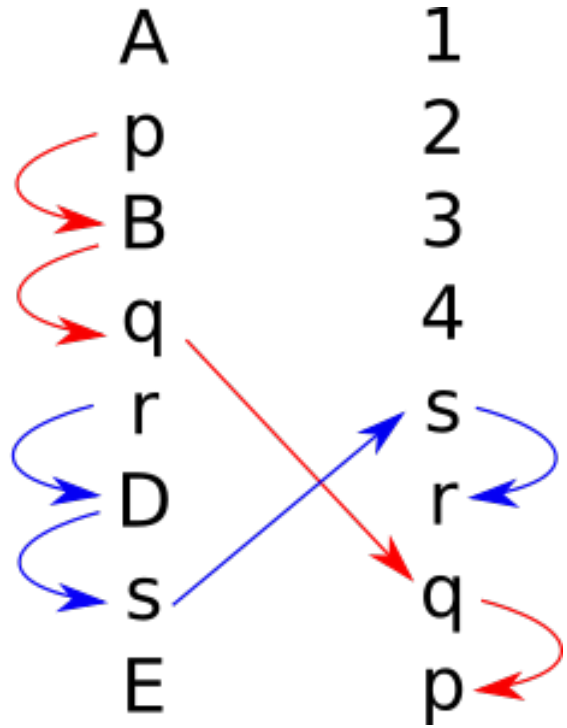
Weiler-Atherton Polygon Clipping



A	1
p	2
B	3
q	4
r	s
D	r
s	q
E	p

Weiler-Atherton Polygon Clipping





Next class

- Time: Sep 21st 9-10
- Topic: Curves