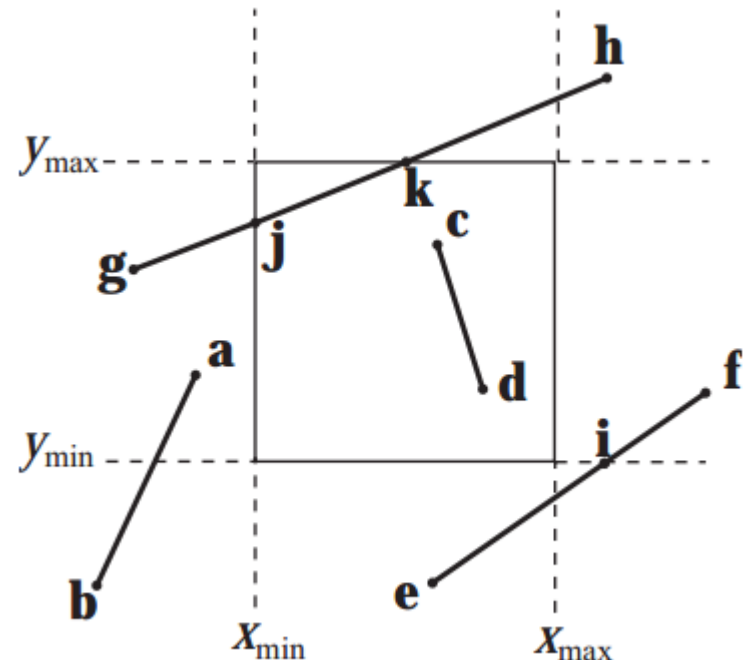


# RECORTE

Prof. Dr. Bianchi Serique Meiguins  
Prof. Dr. Carlos Gustavo Resque dos Santos

# Recorte

- Surgiu da necessidade de não rasterizar elementos que estejam fora da tela (ou do framebuffer)
- Avaliar três casos:
  - ▣ Elemento inteiramente dentro da tela
  - ▣ Elemento inteiramente fora da tela
  - ▣ Elemento possui parte dentro e parte fora da tela



# Vantagens do Recorte

---

- Poupa recursos computacionais
  - ▣ especialmente em aplicações com muitos elementos gráficos que não aparecem em tela.
- Evita problemas com “OutOfBounds”
- Pode ser utilizado aplicado tanto em 2D quanto 3D
- A tela não precisa ser um retângulo (Para alguns algoritmos)

# Tipos de Recorte

---

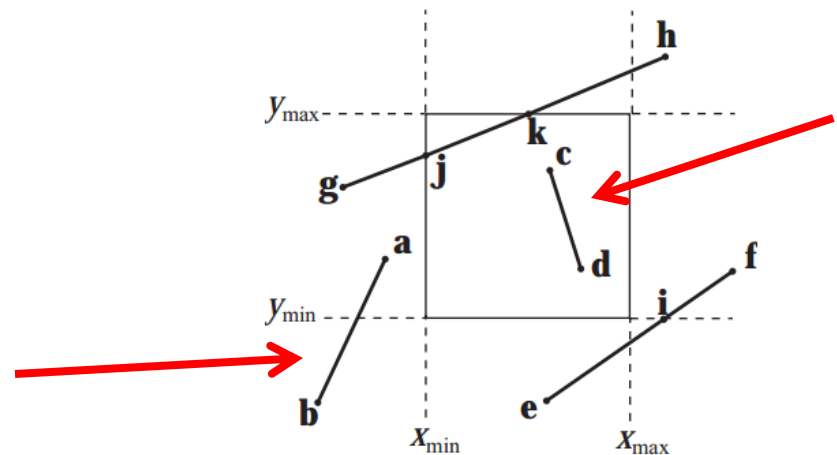
- Recorte de Pontos
- Recorte de Linhas
- Recorte de Polígonos

# Recorte de Pontos

- O mais trivial:
- Basta verificar se o ponto está contido na tela.
- Ou seja:
  - $x_{min} \leq x \leq x_{max} \ E \ y_{min} \leq y \leq y_{max}$

# Recorte de Linhas

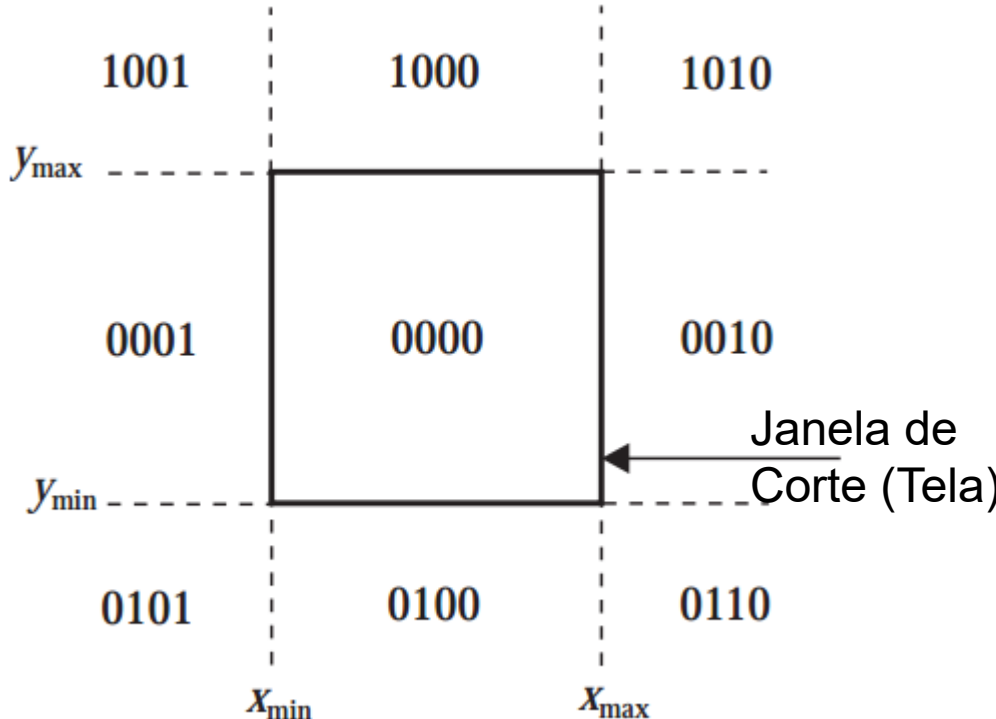
- A primeira vista seria apenas calcular as interseções das linhas com as bordas da tela
- Entretanto temos problemas com cálculos desnecessários nos casos triviais:
  - ▣ Totalmente fora
  - ▣ Totalmente dentro



# Algoritmo de Cohen-Sutherland

- Identifica, de forma eficiente, que linhas são trivialmente aceitas ou rejeitadas
- Dispensa cálculos de interseções nestes casos
- Utiliza operações binárias (apropriado para linguagens de baixo nível)

# Algoritmo de Cohen-Sutherland

- Primeiro passo:
    - Definir um código binário para os pontos da reta
      - $sign(n)$  retorna 0 se  $n \geq 0$  e 1 se  $n < 0$
  - 1º bit:  $sign(y_{max} - y)$
  - 2º bit:  $sign(y - y_{min})$
  - 3º bit:  $sign(x_{max} - x)$
  - 4º bit:  $sign(x - x_{min})$
- 



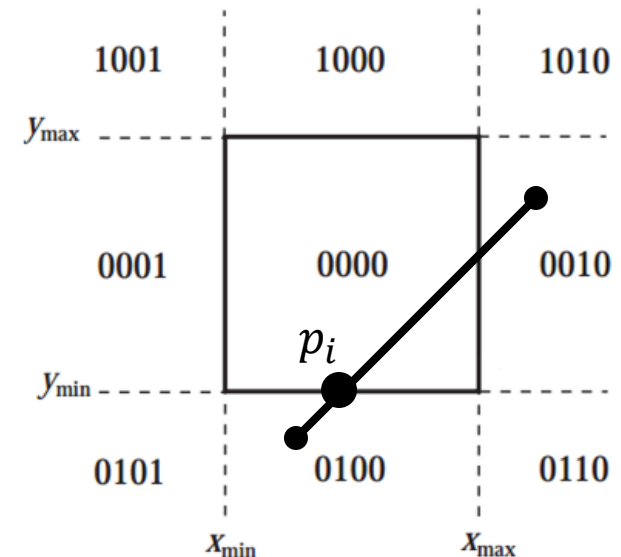
# Algoritmo de Cohen-Sutherland

## □ Segundo Passo:

- Verificar se a linha está totalmente dentro ou totalmente fora
- Considerando  $c_1$  e  $c_2$  os códigos de uma reta
  - Se  $c_1$  OU  $c_2 = 0000$ , então linha está totalmente dentro
  - Se  $c_1$  E  $c_2 \neq 0000$ , então linha está totalmente fora

# Algoritmo de Cohen-Sutherland

- Terceiro Passo:
  - Calcular a interseção ( $p_i$ ) da reta com a linha da tela na qual ocorre a primeira diferença bits dos pontos
  - Considerar  $p_i$  e o ponto que tem zero na posição da diferença como uma nova reta
  - Usar essa reta recursivamente



# Algoritmo de Cohen-Sutherland

- `CS_Clip(p1,p2,xmin,xmax,ymin,ymax):`
  - `int c1 = mkcode(p1) /*gera o código binário para o p1*/`
  - `int c2 = mkcode(p2) /*gera o código binário para o p2*/`
  - `Se ((c1|c2) == 0) desenhaLinha(p1,p2) /*totalmente dentro*/`
  - `Senão Se ((c1&c2) != 0) return null /*totalmente fora*/`
  - `Senão:`
    - `int difBit = findDifBit(p1,p2) /* encontra 1º bit com valor 1 */`
    - `/* calcula a intersecção i entre a reta e a borda do monitor (referente ao bit encontrado anteriormente) */`
    - `Point i = intersect_lines(find_Window_Line(difBit), (p1,p2))`
    - `/* Usa o ponto que tem 0 nesse bit e a intersecção i recursivamente */`
    - `Se (outside(p1, difBit)):`
      - `CS_Clip(i, p2,xmin,xmax,ymin,ymax)`
    - `Else:`
      - `CS_Clip(p1,i,xmin,xmax,ymin,ymax)`

# Algoritmo de subdivisão por ponto médio

- Motivação:
  - ▣ Algoritmo anterior necessitava de cálculos para obtenção da interseção da linha com limites da janela
- Caso particular do Algoritmo anterior, proposto por Sproull e Sutherland

# Algoritmo de subdivisão por ponto médio

- Utilizar apenas soma e divisão por 2 recursivamente.
  - ▣ um teste inicial é aplicado para detectar linhas trivialmente aceitas ou rejeitadas
  - ▣ linhas para as quais o teste inicial falha, são subdivididas em 2 partes iguais

$$\blacksquare x_m = \frac{(x_1 + x_2)}{2}$$

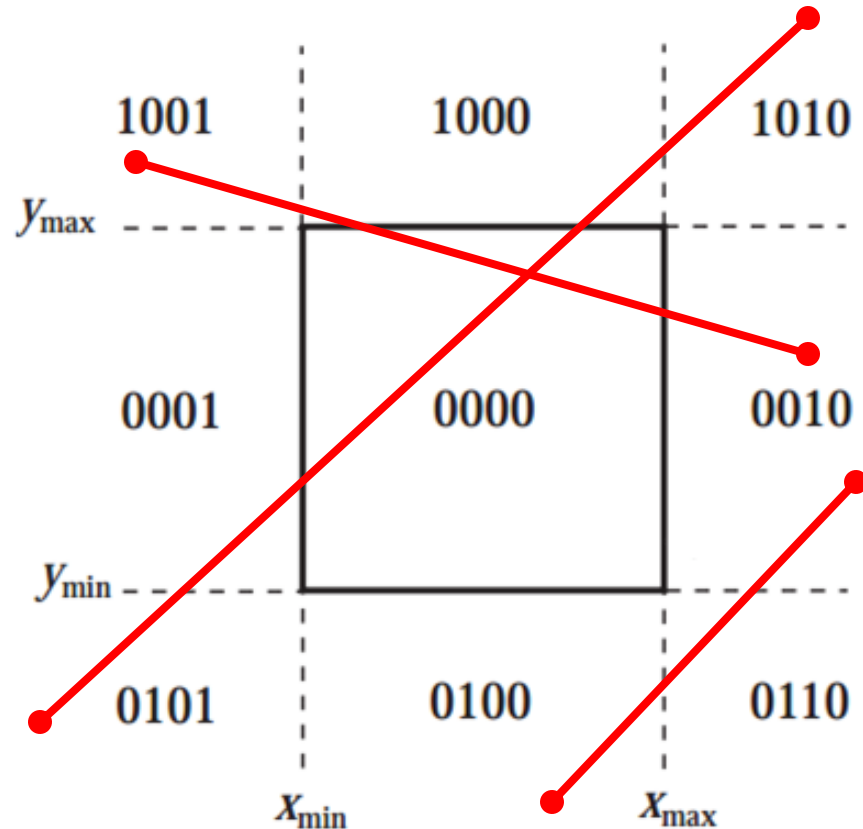
$$\blacksquare y_m = \frac{(y_1 + y_2)}{2}$$

# Algoritmo de subdivisão por ponto médio

- O teste é aplicado nas duas retas resultantes pela divisão  $(p_1, p_m)$  e  $(p_m, p_2)$
- O Teste segue recursivamente:
  - ▣ Aceitando retas totalmente dentro
  - ▣ E rejeitando retas totalmente fora

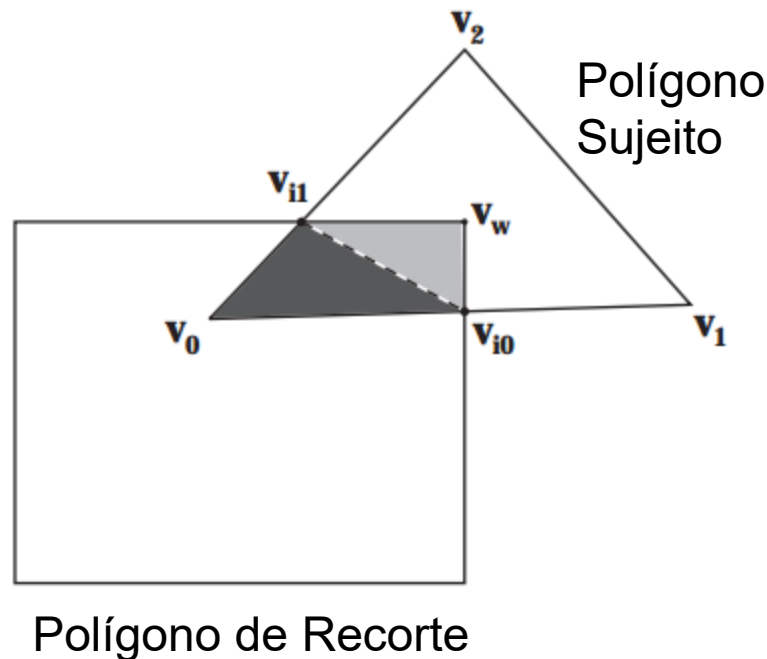
# Algoritmo de subdivisão por ponto médio

## □ Exemplos:



# Recorte de Polígonos

- A priori basta cortar as retas do polígono e ligar os pontos... #sqn :(



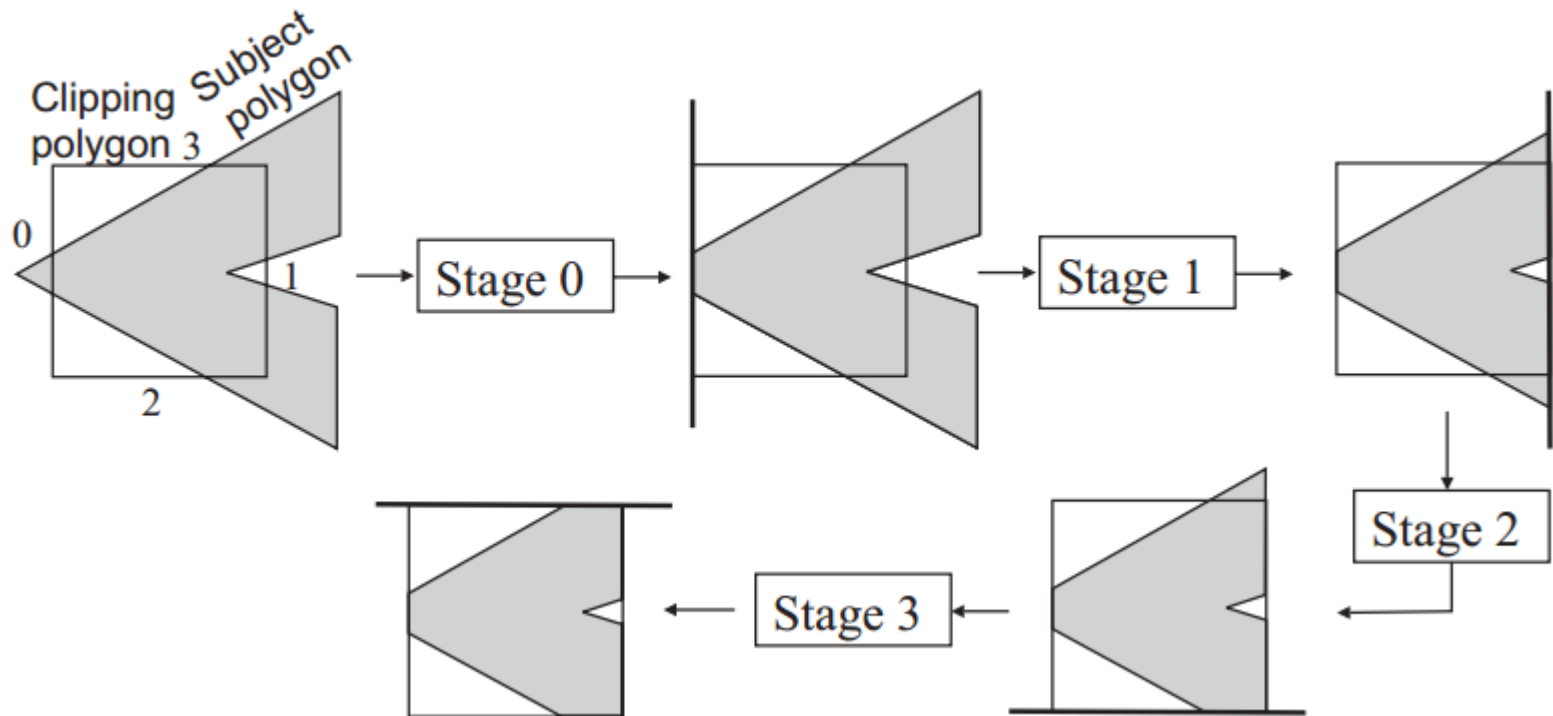


# Algoritmo de Sutherland-Hodgman

- Marco no desenvolvimento da Computação Gráfica
- Só funciona para quando o polígono de recorte é convexo

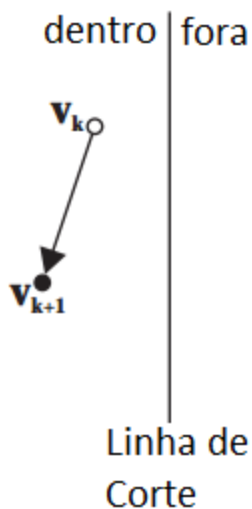
# Algoritmo de Sutherland-Hodgman

- Recortar o Polígono sujeito para cada lado do polígono de recorte

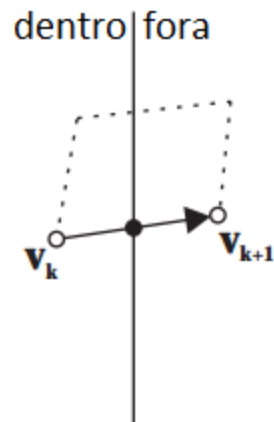


# Algoritmo de Sutherland-Hodgman

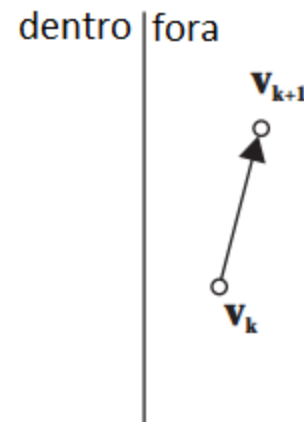
- Casos a serem verificados:
  - Adiciona os vértices de saída em uma lista
  - Essa lista forma o polígono fechado



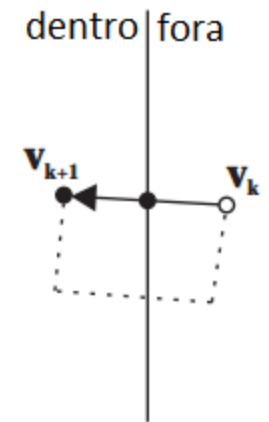
Caso 1: 1 Saída



Caso 2: 1 Saída



Caso 3: 0 Saída



Caso 4: 2 Saídas

● Vértice de Saída