Algoritmos de Preenchimento de Regiões

Sistemas Gráficos/

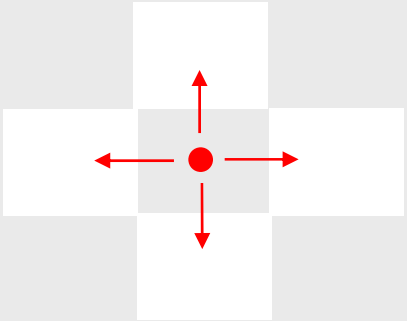
Computação Gráfica e Interfaces

Algoritmos de Preenchimento de Regiões

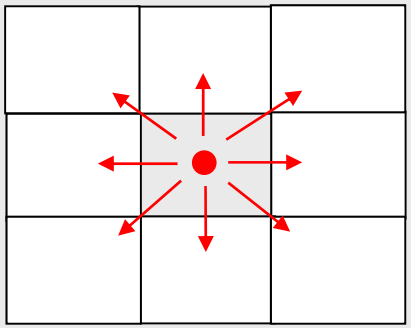
- Classificação dos algoritmos:
  - Preenchimento segundo contorno existente
    - Por difusão [flood-fill]:
      a. Limitado por contorno
      b. Limitado por interior de região
    - Por análise do contorno [boundary algorithm]
  - Preenchimento por varrimento segundo descrição de contorno [scan conversion]
    - Algoritmo da lista de pontos de fronteira ordenados
    - Algoritmo da lista de arestas activas

Algoritmos de Preenchimento de Regiões

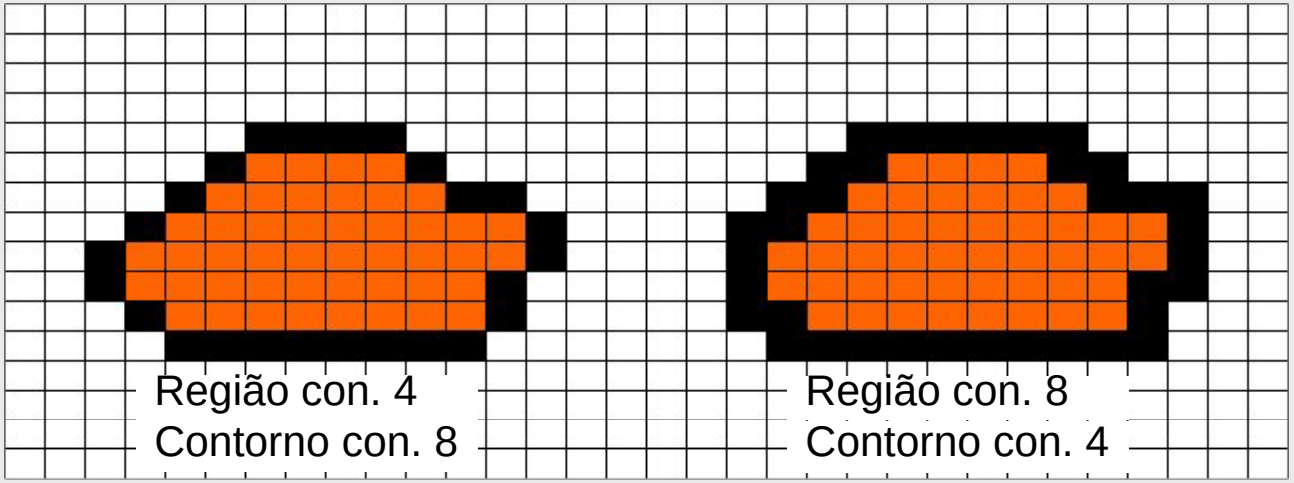
Conectividade 4



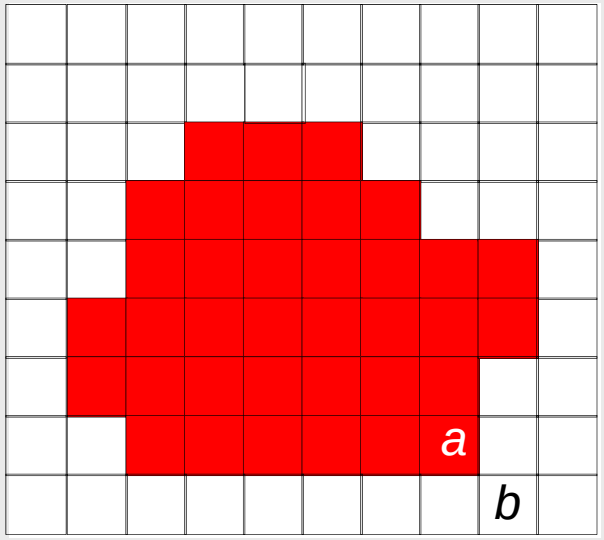
Conectividade 8



Aplicam-se a contorno e região

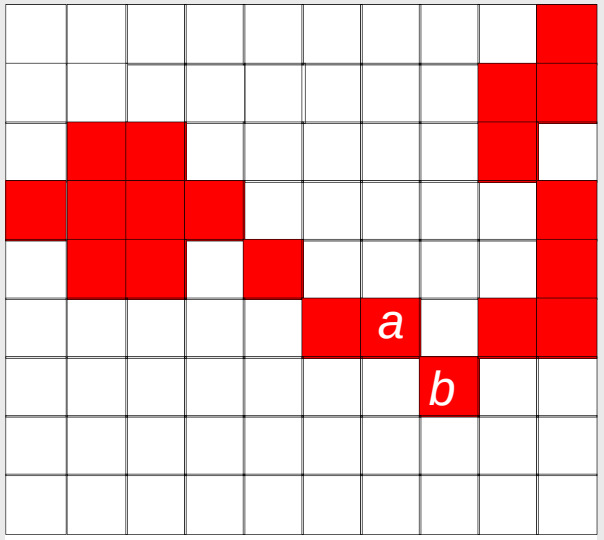


Algoritmos de Preenchimento de Regiões



Região de conectividade 4

b não é vizinho de ***a***



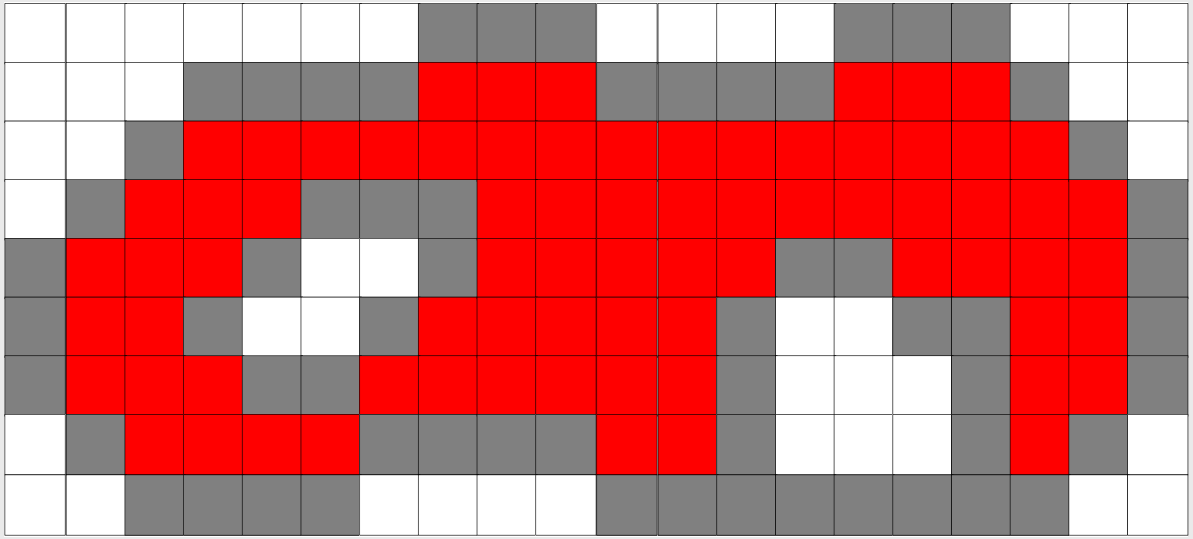
Região de conectividade 8

b é vizinho de ***a***

Algoritmos de Preenchimento de Regiões

Preenchimento segundo contorno existente [flood-fill]

– Limitado pelo contorno



Princípio: Começa num ponto interior e “espalha-se” como se fosse líquido.

Funciona em regiões com buracos.

Algoritmos de Preenchimento de Regiões

Algoritmo para região de conectividade 4:
(contorno pode ser de conectividade 4 ou 8)

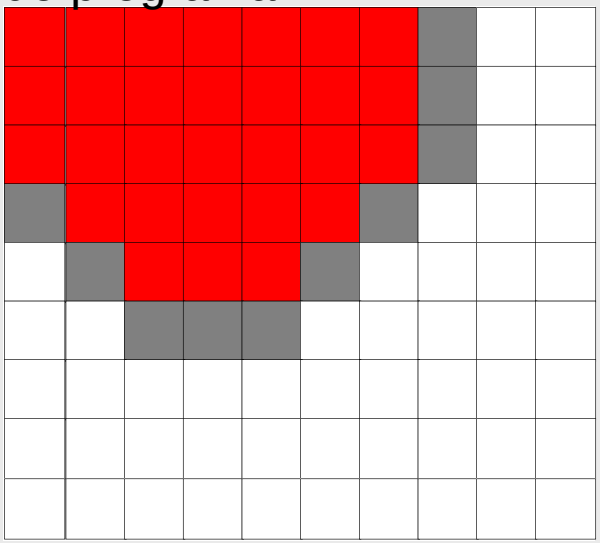
```
void floodfill(int x, int y)
{ if (pointColor(x,y) <> ContourColor && (pointColor(x,y) <> FillColor)
   { ChangeColor(x,y, FillColor);
     // apelo recursivo aos 4 vizinhos
     floodfill(x+1,y);
     floodfill(x-1,y);
     floodfill(x,y+1);
     floodfill(x,y-1);
   }
}
```

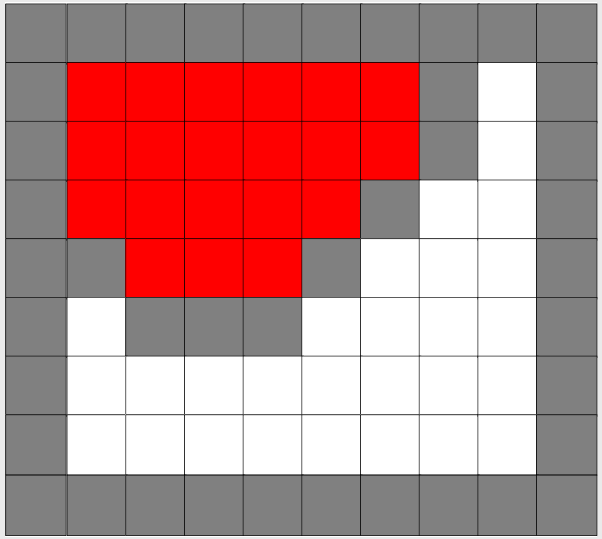
Para região de connectividade 8: chama recursivamente a função floodfill para os oito vizinhos. Para além dos indicados temos:

(x+1, y+1), (x-1, y+1), (x-1, y-1), (x+1, y-1)

Algoritmos de Preenchimento de Regiões

Fronteira não completamente fechada → pode originar erro durante a execução do programa





Evitam-se os erros se a leitura pointColor(x,y) fornecer o valor correspondente a ContourColor no caso do ponto se encontrar fora do ecrã.

Algoritmos de Preenchimento de Regiões

Preenchimento segundo contorno existente [flood-fill]

– **região definida pelo seu interior**

```
void floodfill(int x, int y)
{  if (pointColor(x,y) == RegionColor)
   {    ChangeColor(x,y, FillColor);
        // apelo recursivo aos 4 vizinhos
        floodfill(x+1,y);
        floodfill(x-1,y);
        floodfill(x,y+1);
        floodfill(x,y-1);
   }
}
```

Aplicação: para substituir uma cor por outra

Problemas: consumo de *stack* (pilha)

Soluções para minimizar o tamanho da ***stack***:

- Evitar declarar variáveis locais
- Não passar a cor de preenchimento como parâmetro

Notar que agora não existe o problema da fronteira incompleta.

Algoritmos de Preenchimento de Regiões

- Classificação dos algoritmos:
  - Preenchimento segundo contorno existente
    - Por difusão [flood-fill]:
      a. Limitado por contorno
      b. Limitado por interior de região
    - **Por análise do contorno [boundary algorithm]**
  - Preenchimento por varrimento segundo descrição de contorno [scan conversion]
    - Algoritmo da lista de pontos de fronteira ordenados
    - Algoritmo da lista de arestas activas

Algoritmos de Preenchimento de Regiões

Preenchimento segundo contorno existente

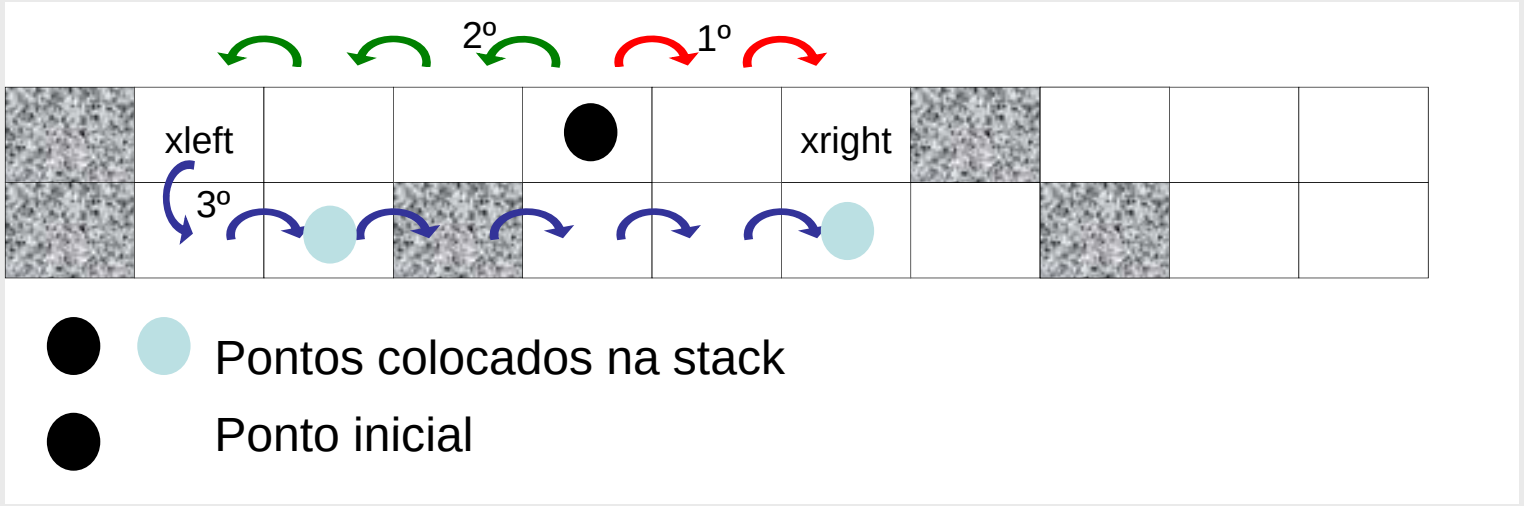
– Por análise do contorno [boundary algorithm]

Princípio: trabalha linha a linha e apenas coloca na pilha algumas extremidades de segmentos.

Algoritmo:

1. Parte de um ponto inicial, situado no interior, que começa por ser colocado na pilha.
2. Se pilha vazia, então termina,
   senão retira um ponto da pilha.
3. A partir desse ponto preenche na horizontal, para a direita e, em seguida, para a esquerda até encontrar o contorno. Toma nota das extremidades Xleft e Xright.
4. Na linha imediatamente abaixo procura, entre Xleft e Xright, os novos pontos de partida. Estes pontos são colocados na pilha.
5. Idem 4, para a linha imediatamente acima.
6. Volta a 2.

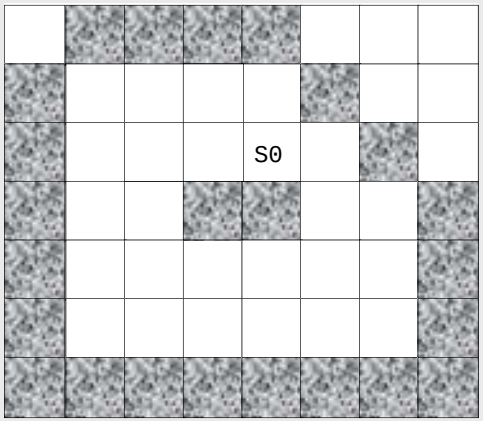
Preencimento de regiões por análise do contorno

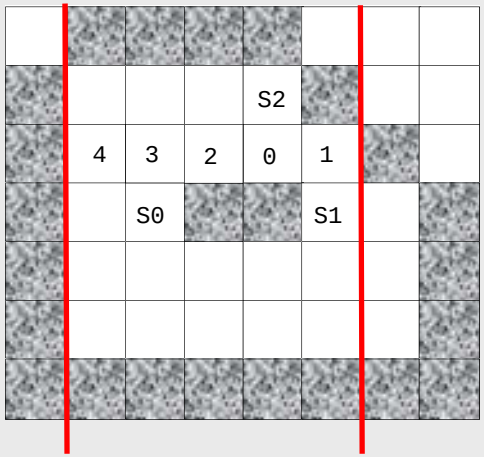


Preencimento de regiões por análise do contorno

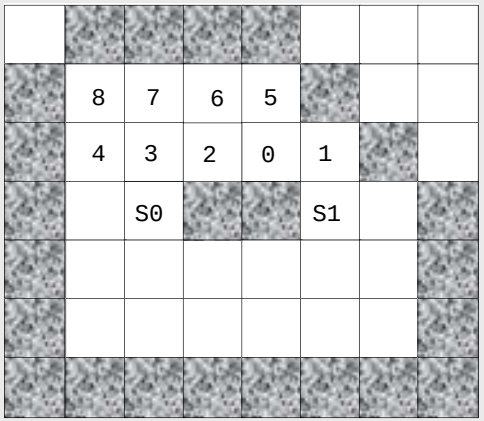
Exemplo:

Processa s0

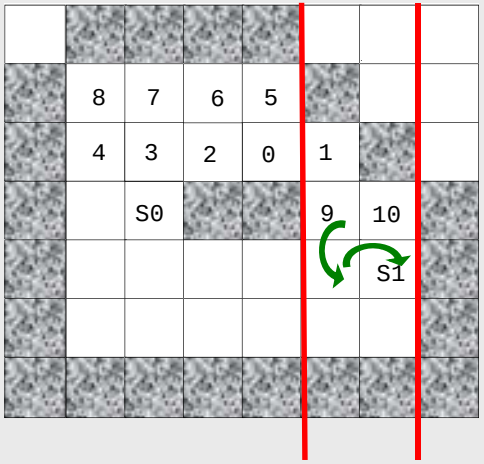




Processa s2



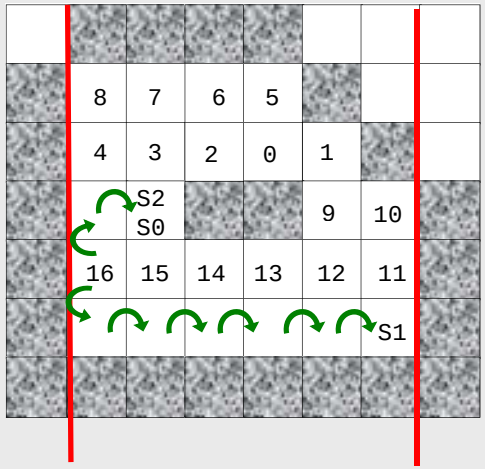
Processa s1



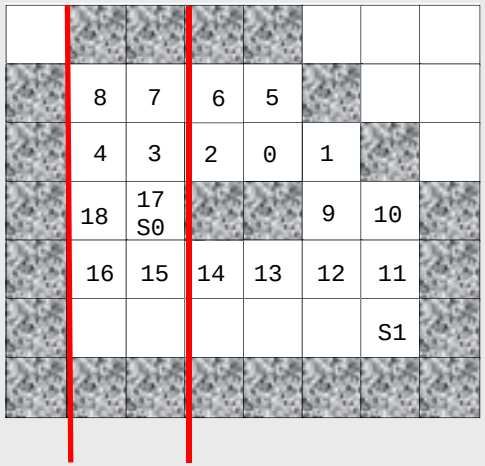
Ponto seguinte ?

Preencimento de regiões por análise do contorno

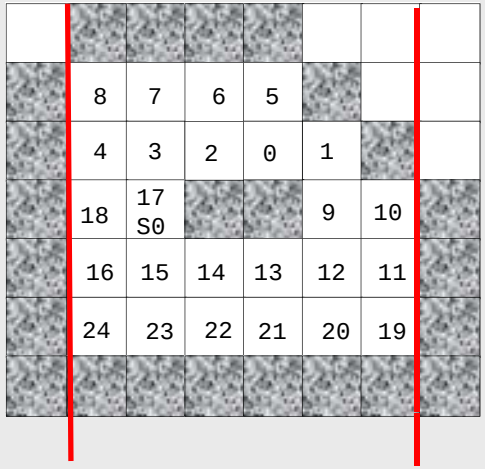
Processa s1



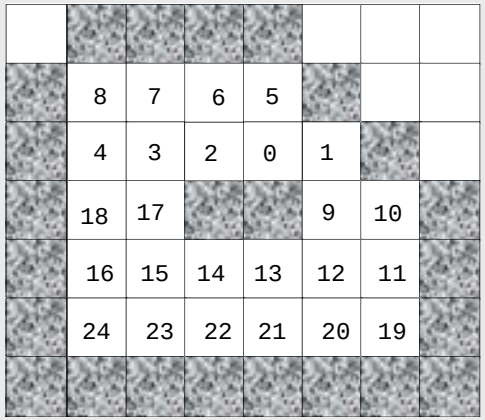
Processa s2



Processa s1



Processa s0



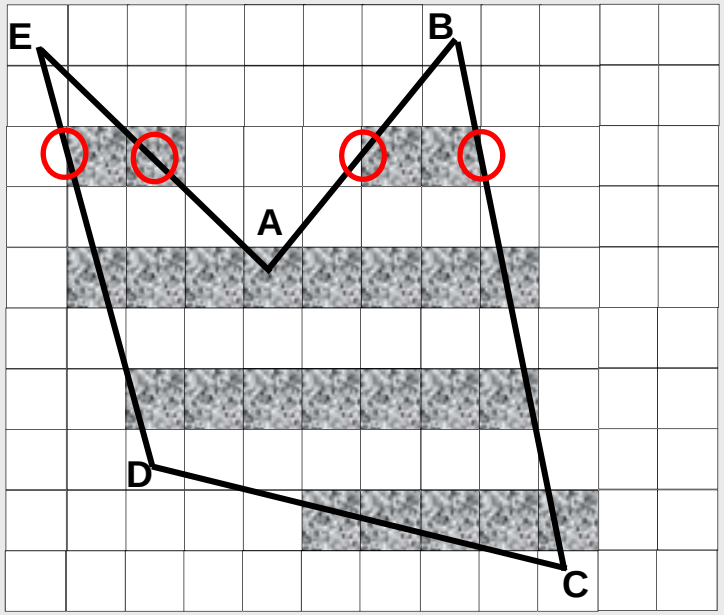
Pilha vazia - Fim

Algoritmos de Preenchimento de Regiões

- Classificação dos algoritmos:
  - Preenchimento segundo contorno existente
    - ➢ Por difusão [flood-fill]:
      a. Limitado por contorno
      b. Limitado por interior de região
    - ➢ Por análise do contorno [boundary algorithm]
  - **Preenchimento por varrimento segundo descrição de contorno [scan conversion]**
    - ➢ Algoritmo da lista de pontos de fronteira ordenados
    - ➢ Algoritmo da lista de arestas activas

Algoritmos de Preenchimento de Regiões

Preenchimento por varrimento segundo descrição de contorno [scan conversion] - Algoritmo da lista de pontos de fronteira ordenados



O algoritmo determina as intersecções das arestas com as linhas de varrimento do ecrã e ordena-as. Os pontos a preencher estão entre dois pares de pontos.

Algoritmos de Preenchimento de Regiões

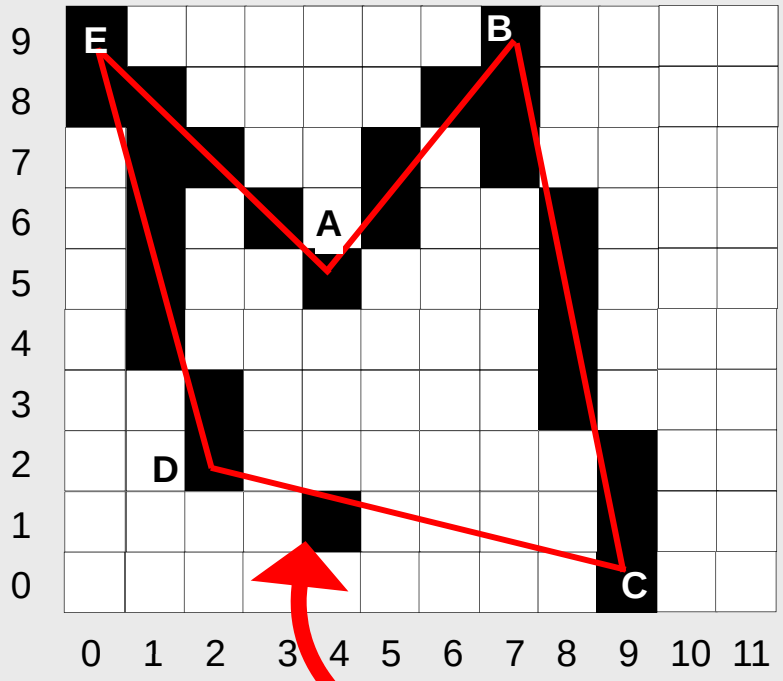
Algoritmo:

1. Determinação das intersecções das arestas com as linhas de varrimento do ecrã (utilizando, por exemplo, o algoritmo MidPoint modificado, de tal forma que produza um só ponto por horizontal). ← Designados de pontos de fronteira
2. Ordenação dos pontos obtidos. Primeiro segundo Y e, em seguida, para o mesmo Y, segundo X.

   X1,Y1 precede X2,Y2 se Y1 < Y2 ou

   se Y1 = Y2 e X1 <=X2
3. Os segmentos horizontais de preenchimento são agora especificados considerando pares de pontos consecutivos.

Algoritmos de Preenchimento de Regiões



Pontos de fronteira obtidos (por linha):

(0,9) (0,9)	(7,9) (7,9)
(0,8) (1,8)	(6,8) (7,8)
(1,7) (2,7)	(5,7) (7,7)
(1,6) (3,6)	(5,6) (8,6)
(1,5) (4,5)	(4,5) (8,5)
(1,4) (8,4)	
(2,3) (8,3)	
(2,2) (9,2)	
(4,1) (9,1)	
(9,0) (9,0)	

Cuidado com os vértices duplos

Esta aresta só gera um ponto de intersecção

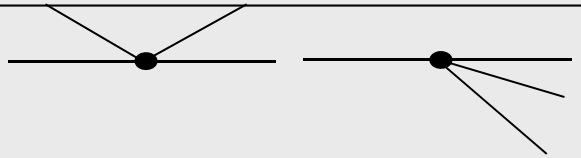
Simplificação da estrutura de dados: guardar por segmentos (x1, x2, y).

Ex: (0,0,9) (7,7,9)

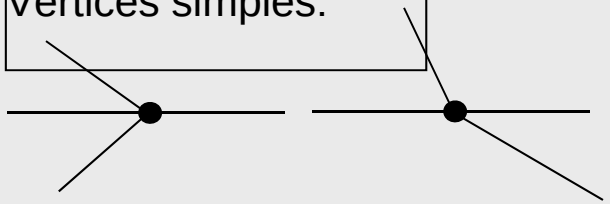
(0,1,8) (6,7,8) ....

Algoritmos de Preenchimento de Regiões

Os vértices duplos podem causar problemas:



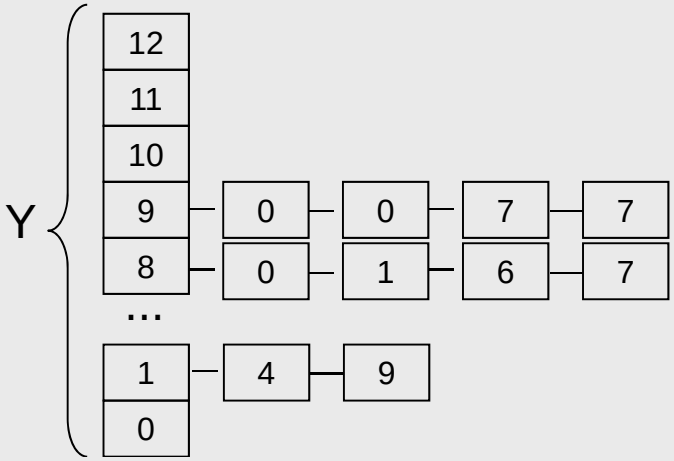
Vértices simples:



Desvantagem do algoritmo: a ordenação pode ser um processo lento por envolver um elevado número de pontos.

Melhoramento: **Algoritmo da tabela de listas de pontos ordenados**

Consiste em construir uma lista ordenada de pontos para cada valor de Y.

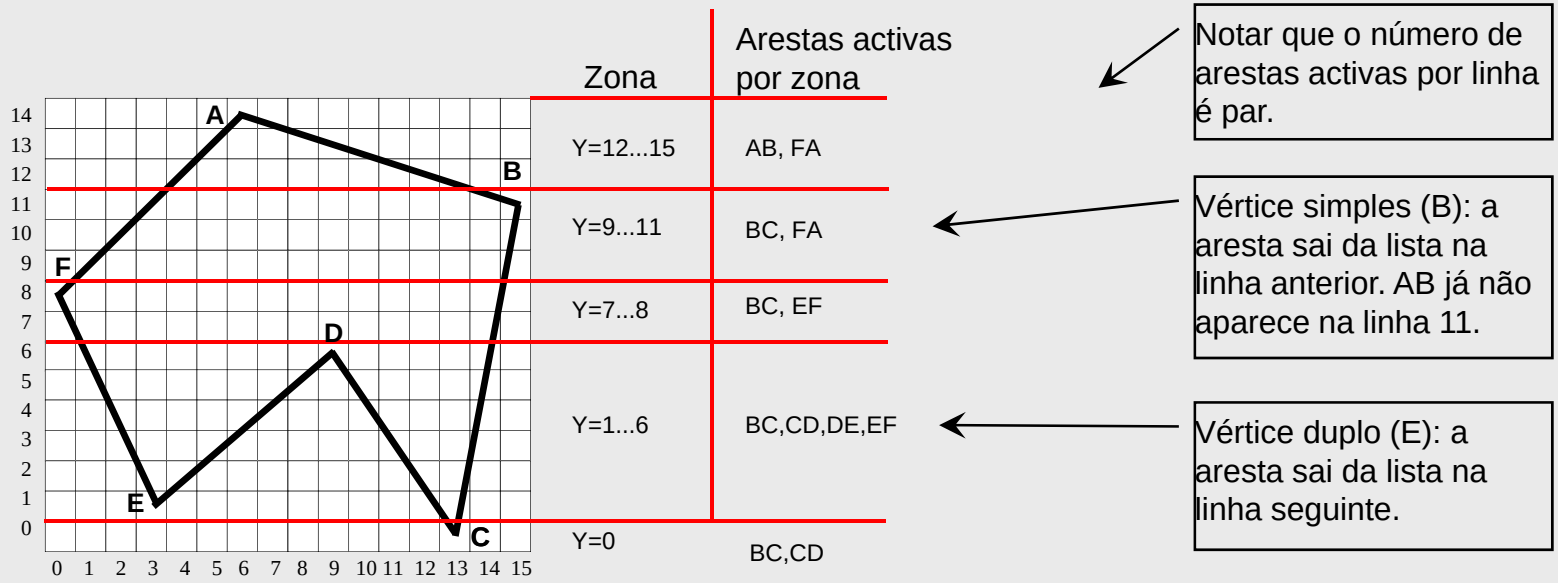


Algoritmo:

1. Determinar as intersecções (x_i,y_i) para cada aresta. Para cada intersecção colocar x_i na lista y_i.
2. Em cada lista y_i, ordenar os valores X por ordem crescente.
3. Em cada lista y_i, considerar os pares de valores X consecutivos, que definem os segmentos horizontais a visualizar.

Algoritmos de Preenchimento de Regiões

Preenchimento por varrimento segundo descrição de contorno [scan conversion] - Algoritmo da lista das arestas activas



- O preenchimento realiza-se por linha de varrimento do ecrã, pelo que será viável tratar e memorizar apenas os pontos relativos a essa linha.

- Só as arestas activas entram no preenchimento de uma linha de varrimento.

- → Interessa manter a **Lista das Arestas Activas**.

- Esta lista é actualizada sempre que se entra numa nova zona.

Algoritmos de Preenchimento de Regiões - Algoritmo da lista das arestas activas

Algoritmo da lista das arestas activas :

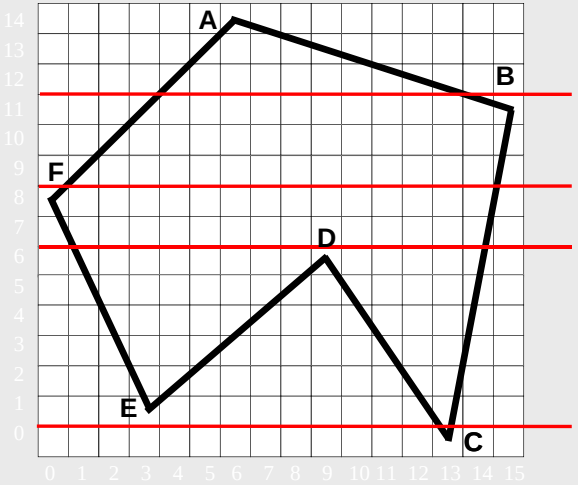
1. Constituição da **Tabela das Arestas**
    - Para cada aresta é memorizado:
        - A coordenada X.
        - DX, valor a adicionar a X, para encontrar o ponto seguinte quando se incrementa Y de 1.
        - LongY, comprimento da aresta segundo o eixo Y.
2. Para cada linha de varrimento:
    - Verificar na tabela de arestas se existem novas arestas nesta linha. Em caso afirmativo, juntá-las à Lista das Arestas Activas.
    - Ordenar os valores de X.
3. Agrupa aos pares, os valores de X que definirão os segmentos horizontais a visualizar.
4. No final da linha preparar a informação para a linha seguinte:

    Para cada Aresta Activa:

    Decrementar o valor LongY. Se LongY=0, então a aresta respectiva sai da lista das arestas activas, senão é calculado o novo X, adicionando DX ao valor actual.
5. Voltar a 2.

Algoritmos de Preenchimento de Regiões -
Algoritmo da lista das arestas activas

O primeiro passo do algoritmo será a classificação dos **vértices** em: **simples** ou **duplos**. A seguir constrói-se a tabela das arestas.



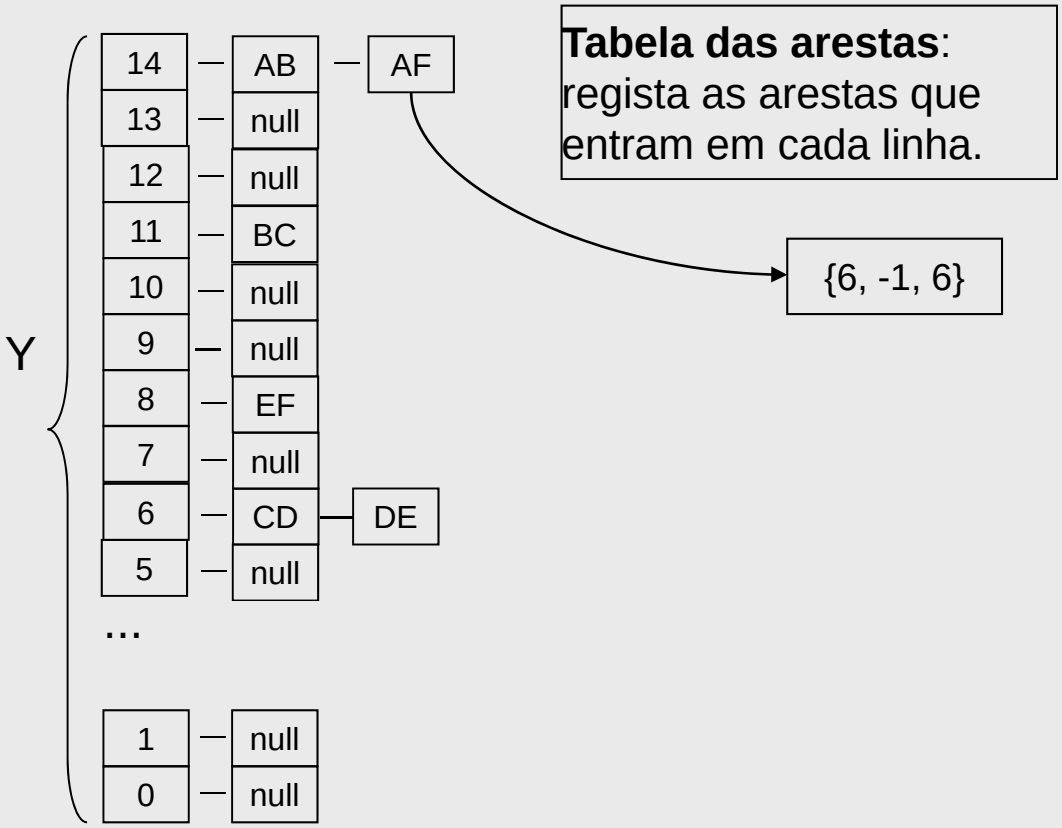


Tabela das arestas: regista as arestas que entram em cada linha.

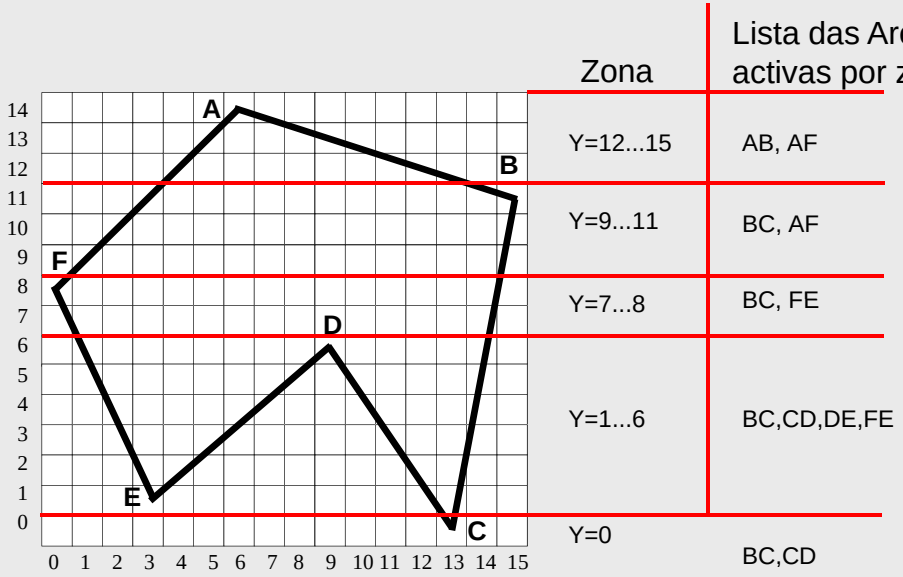
Algoritmos de Preenchimento de Regiões -
Algoritmo da lista das arestas activas

A análise é efectuada de cima para baixo e da direita para a esquerda.

Se vértice simples: longY = y2 – y1

Se vértice duplo: longY = y2 – y1 + 1

Algoritmos de Preenchimento de Regiões



Zona	Lista das Arestas activas por zona
Y=12...15	AB, AF
Y=9...11	BC, AF
Y=7...8	BC, FE
Y=1...6	BC,CD,DE,FE
Y=0	BC,CD

{X, DX, LongY}

AB → {6, 3, 3}

BC → {15, -0.18, 12}

CD → {9, 0.66, 7}

DE → {9, -1.2, 6}

FE → {0, 0.43, 8}

AF → {6, -1, 6}

1º Passo Y=14

Lista = {AB,AF}

Pares de valores X: (6,6)

AB→(9,3,2) AF→(5,-1,5)

2º Passo Y=13

Lista = {AB,AF}

Pares de valores X: (5,9)

AB→(12,3,1) AF→(4,-1,4)

3º Passo Y=12

Lista = {AB,AF}

Pares de valores X: (4,12)

AB→~~(15,3,~~**0**) AF→(3,-1, 3)

4º Passo Y=11

Lista = {BC,AF}

Pares de valores X: (4,15)

BC→(14.82,-0.18,10) AF→(2,-1, 2)

...

Exercício

5. Seja um polígono definido pela sucessão de vértices {(1,6), (6,2), (6,6)} a ser preenchido pelo algoritmo da lista de pontos de fronteira ordenados.

a) Apresente o resultado dos dois passos iniciais do algoritmo, quando aplicado ao polígono em questão.

b) Explique como se efectua o preenchimento do polígono, com base nos resultados da alínea anterior.

Exame de 20 de Junho de 2002

6. Seja um polígono fechado, definido pela sucessão de vértices seguinte, a ser preenchido pelo algoritmo da Lista de Arestas Activas.

{(5, 1), (2, 4), (4, 6), (9, 6), (11, 4), (8, 1), (8, 4), (6, 2), (5, 3)}

a)- Mostre qual é o conteúdo da tabela de arestas inicial.

b)- Mostre qual é o estado da lista de arestas activas AEL nas linhas de varrimento 2, 3 e 4, logo após a inserção das novas arestas respectivas.

(Exame de 13 de Julho de 2002)