

BÀI GIẢNG ĐỒ HỌA MÁY TÍNH

THUẬT GIẢI TÔ MÀU

NGÔ QUỐC VIỆT

2010

Nội dung

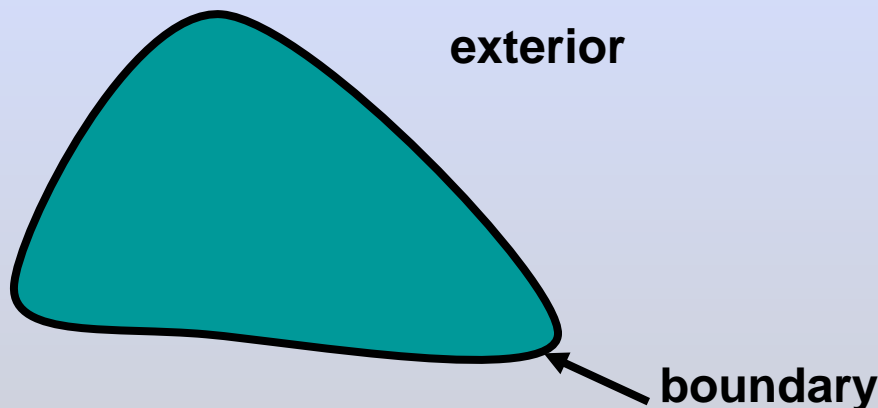
- Giới thiệu.
- Thuật giải tô màu theo đường quét.
- Thuật giải dầu loang.
- Giải đáp thắc mắc
- Bài tập

Giới thiệu

- Tô vùng trong của một bề mặt trên thiết bị raster. Cụ thể, tô đa giác (vì có thể xấp xỉ bề mặt bởi tập các đa giác).
- Tô màu đặc hay mẫu tô bất kỳ.
- Tận dụng kết quả vẽ đoạn thẳng giữa hai điểm.
- Sử dụng các kỹ thuật khác?

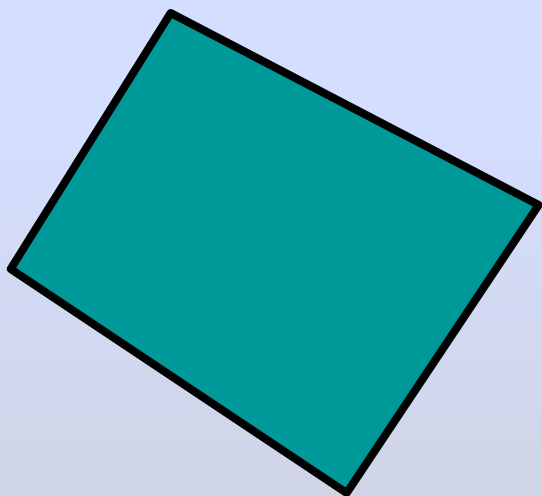
Tô màu theo đường quét

- Vùng được định nghĩa bởi màu của pixel, chia làm 3 phần:
 - Vùng trong – interior
 - Vùng ngoài – exterior
 - Biên (liên tục) - boundary

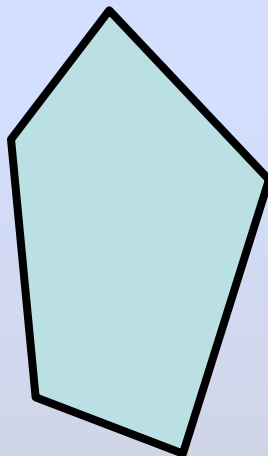


Tô màu theo đường quét

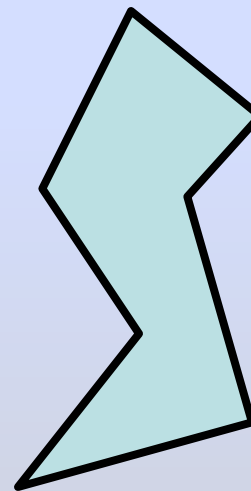
- Định nghĩa bằng đa giác: xác định các đỉnh các đỉnh $p_i = (x_i, y_i)$
- Các loại đa giác: Convex; Concave; Simple; Nonsimple



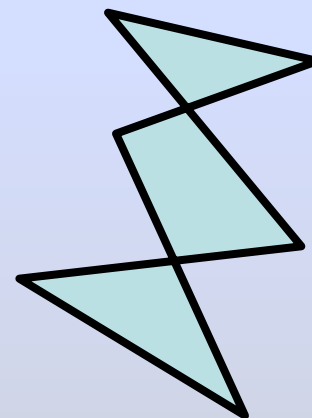
polygonal region



convex



concave

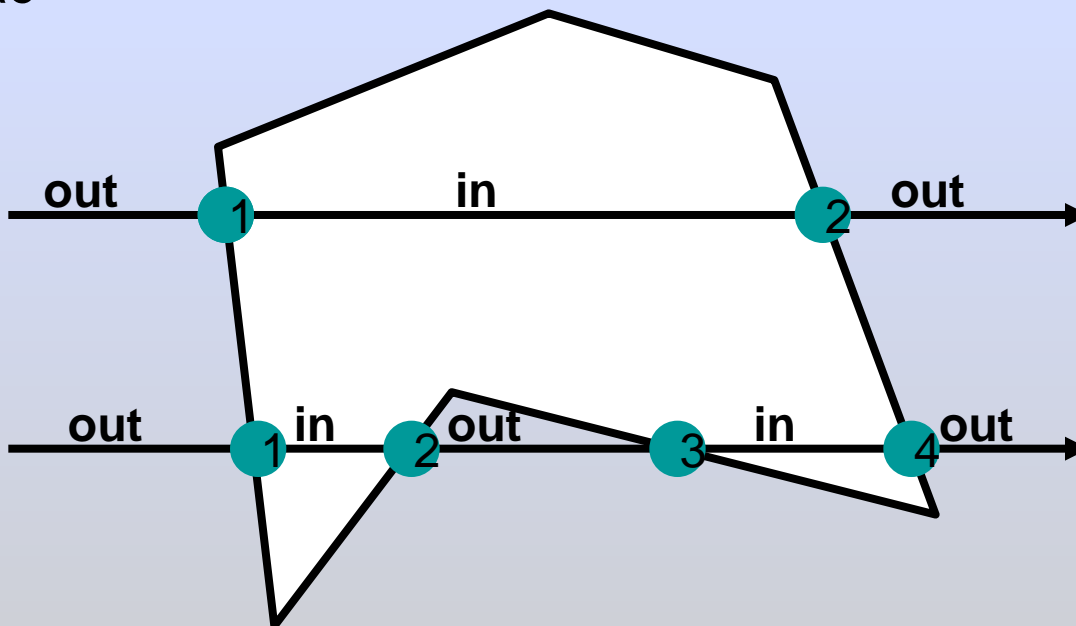


nonsimple

Tô màu theo đường quét

Scanline

- Đường thẳng nằm ngang
- Số giao điểm của scanline và đa giác là số chẵn (tổng quát)
- Các pixel nằm giữa các cặp giao điểm lẻ-chẵn nằm trong đa giác



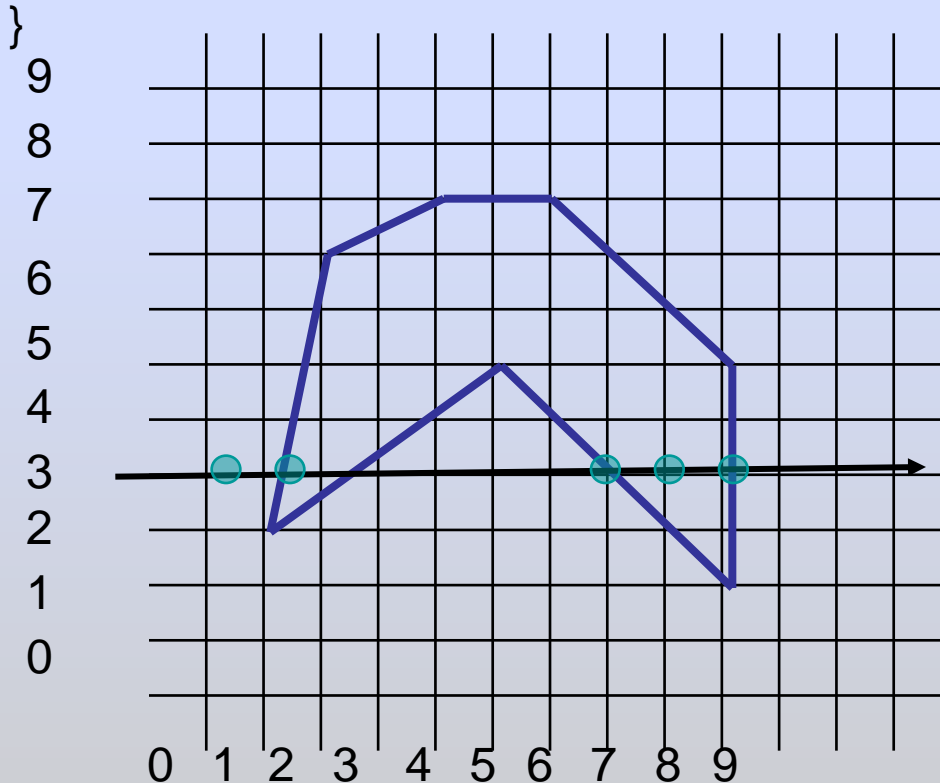
Scanline tổng quát

for each scanline {

 Tìm giao điểm của scanline với các cạnh của đa giác

 Sắp xếp các giao điểm theo thứ tự tăng dần theo x

 Tô các pixel nằm giữa các cặp giao điểm liên tiếp nhau

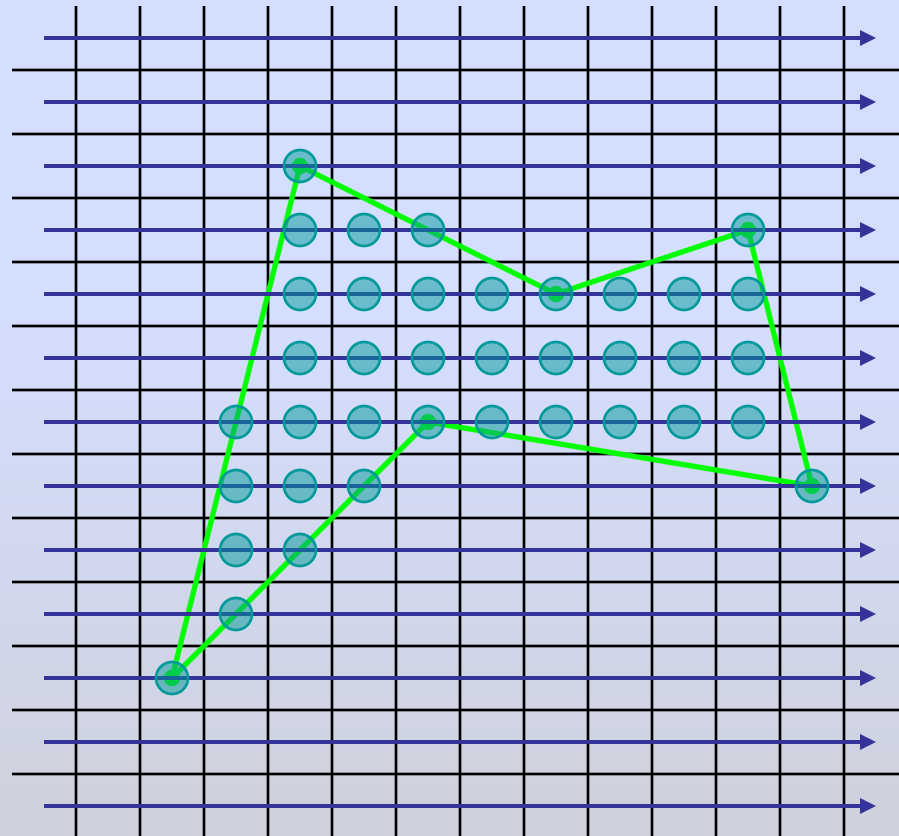


Tại dòng scanline $y = 3$:

Các hoành độ giao điểm sau khi làm tròn là 1, 2, 7, 9

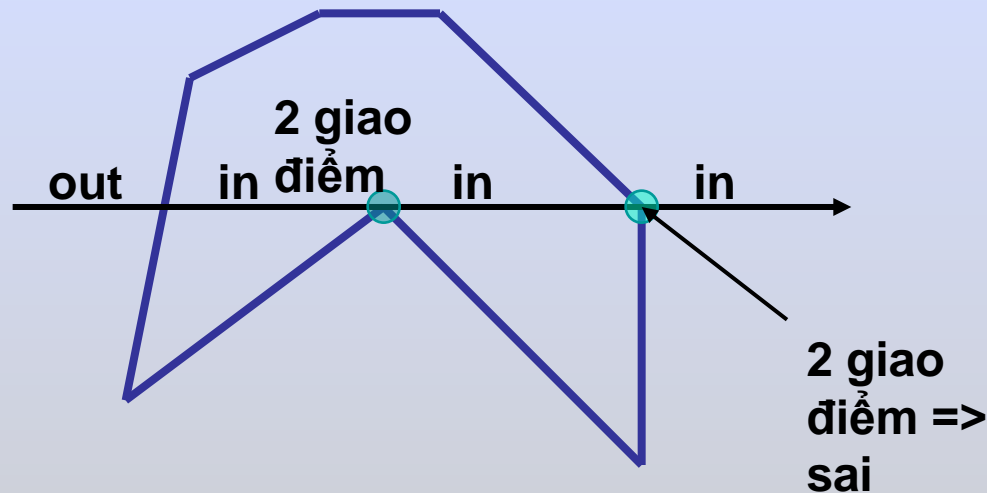
Do đó, 2 đoạn [1,2] và [7,9] được tô

Thuật giải scanline tổng quát



Scanline-Các trường hợp đặc biệt

- Các cạnh nằm ngang không xét đến vì chúng sẽ được tô khi xét 2 cạnh kề với nó.
- Khi scanline đi qua đỉnh của đa giác, nó sẽ giao với 2 cạnh. Trong trường hợp đỉnh không là cực trị, số giao điểm của scanline với đa giác là **số lẻ**.

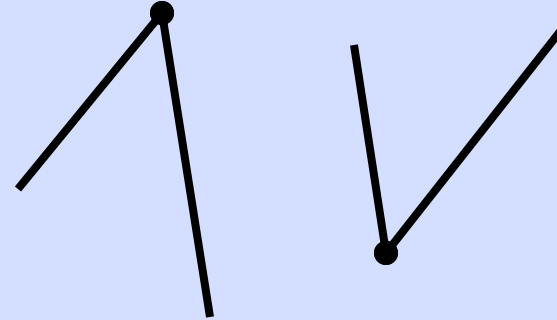


Scanline-Các trường hợp đặc biệt

Các đỉnh cực trị:

minimum

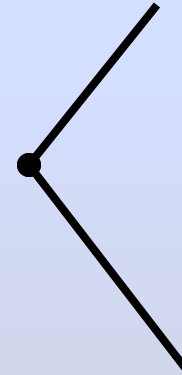
maximum



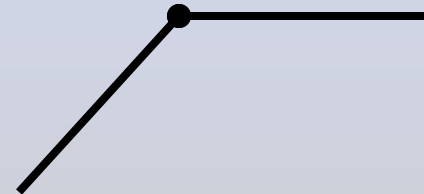
Tăng theo y:

minimum của 1 cạnh

maximum của cạnh còn lại



Cạnh nằm ngang

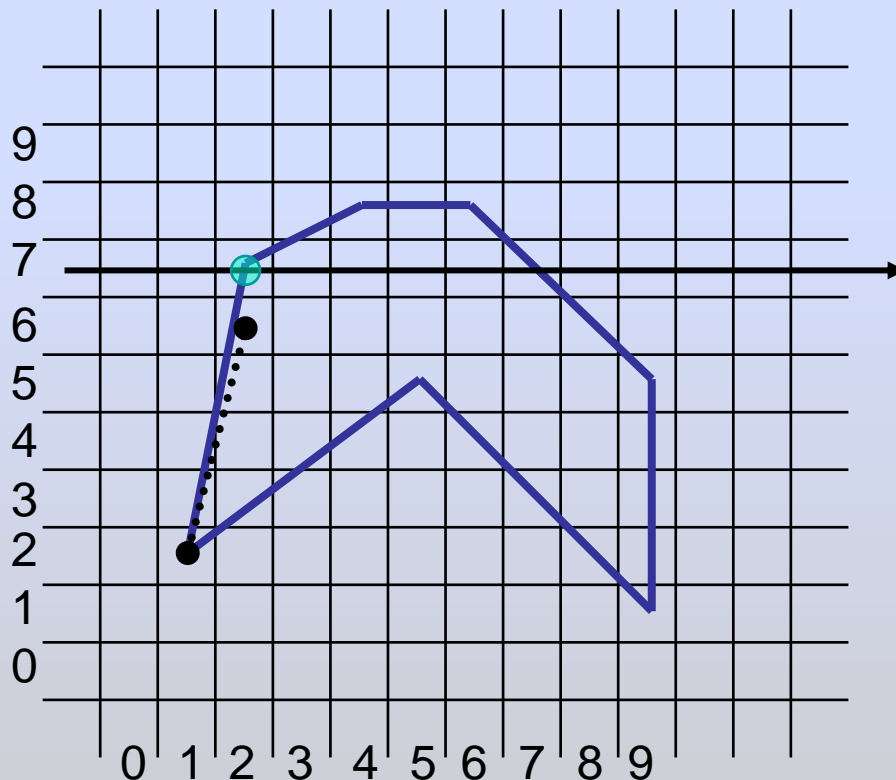


Scanline-Các trường hợp đặc biệt

Trước quá trình tô màu, kiểm tra các đỉnh.

Nếu đỉnh không phải là cực trị, xét cạnh phía dưới.

Giảm tung độ trên y_upper xuống một đơn vị



Danh sách đỉnh đa giác trước khi cải tiến:

$(6,8), (9,5), (9,1), (5,5), (1,2), (2,7), (4,8)$

Sau khi cải tiến, danh sách các cạnh của đa giác như sau - một cạnh bị xóa và 2 cạnh được rút gọn:

$e_1 = (6,8) \text{ to } (9,5)$

$e_2 = (9,4) \text{ to } (9,1)$

$e_3 = (9,1) \text{ to } (5,5)$

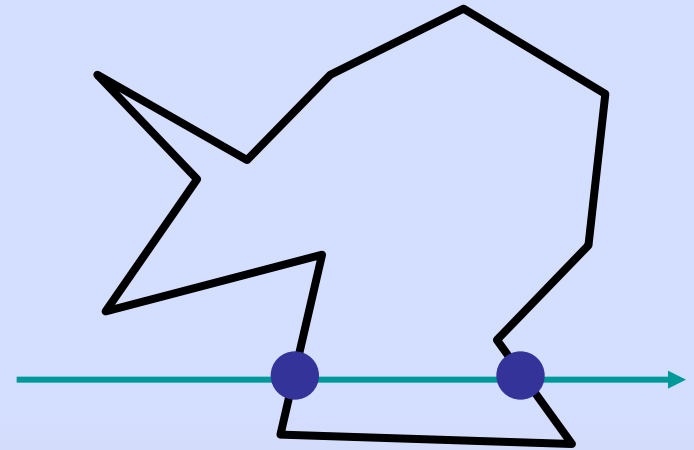
$e_4 = (5,5) \text{ to } (1,2)$

$e_5 = (1,2) \text{ to } (2,6)$

$e_6 = (2,7) \text{ to } (4,8)$

Nhược điểm của scan line tổng quát

- Để xác định giao điểm của đường scanline và cạnh của đa giác, cần phải duyệt tất cả các cạnh của đa giác.
- Khi số cạnh của đa giác lớn, phải mất rất nhiều thời gian để duyệt hết các cạnh, trong khi số cạnh mà đường scanline cắt thì rất ít.
- Chưa thừa kế kết quả của bước trước.

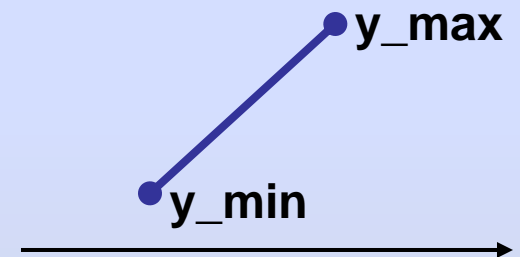
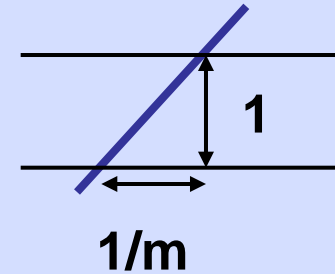


Số giao điểm là 2, trong khi số cạnh là 12

Scan line-cải tiến tốc độ

Nhận xét:

- Khi dòng quét tăng một đơn vị theo y thì hoành độ giao điểm thay đổi theo $1/m$
- Giả sử rằng 1 cạnh của đa giác có tung độ bị chặn bởi $[y_{\min}, y_{\max}]$ thì khi tung độ của dòng quét không thuộc đoạn này, chúng không cắt cạnh đó

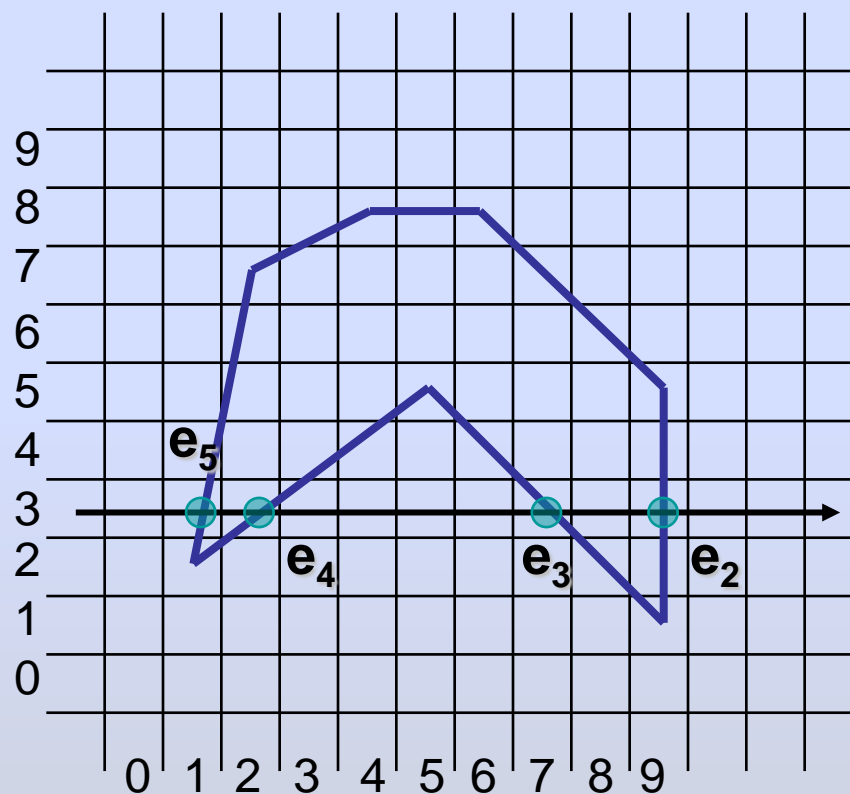


Scan line-sử dụng AEL

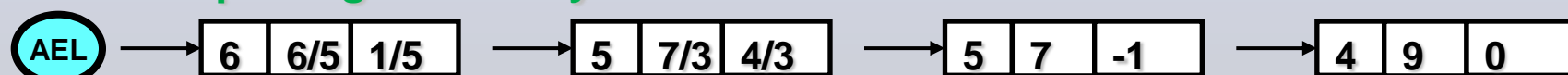
- Để gia tăng tốc độ tính toán, chúng ta xây dựng và duy trì một danh sách xác định tọa độ giao điểm của đa giác và đường scanline ở mỗi bước (AEL).
- Danh sách này cho phép tính toán giao điểm một cách nhanh chóng bằng cách lưu các thông tin các cạnh mà đường scanline cắt.
- Để thuận lợi tính toán, một cạnh có các thông tin sau:
 - Tung độ cao nhất **y_upper** của cạnh (sau khi rút gọn).
 - Hoành độ giao điểm **x_intersection** với đường scanline hiện hành.
 - Nghịch đảo hệ số góc 1/m : Chú ý, 1/m được tính trước khi cạnh được rút gọn, do đó bảo đảm tính chính xác của giao điểm.

y_upper	x_int	recip_slope
----------------	--------------	--------------------

Sử dụng AEL



Tại dòng scanline $y = 3$:



y_{upper} x_{int} $1/m$ Đồ họa máy tính-Ngô Quốc Việt

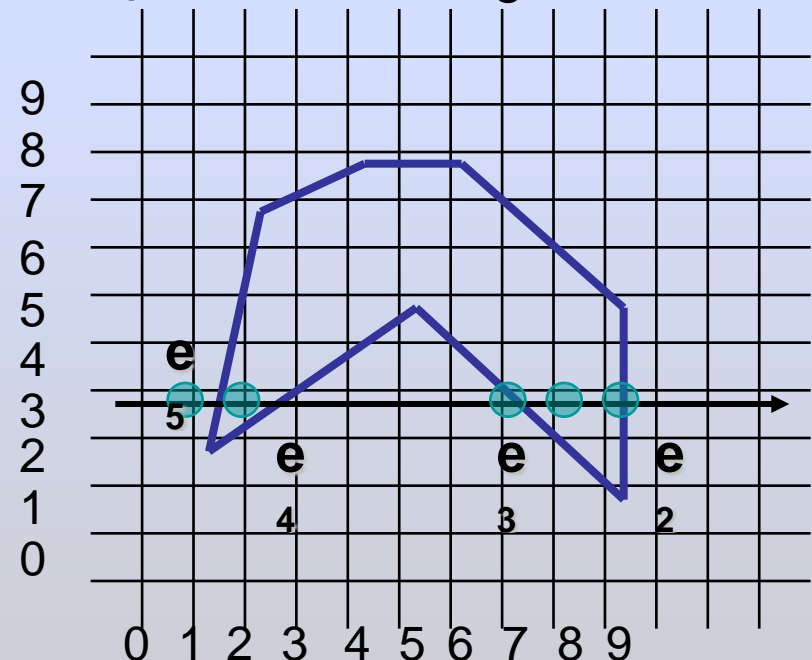
Tô màu tại một dòng quét

Tại dòng scanline hiện hành y , AEL lưu trữ giao điểm của scanline và cạnh đa giác.

Để tô màu, chúng ta sắp xếp các cạnh theo chiều tăng dần của hoành độ giao điểm x_int .

Mỗi cặp giá trị của x_int xác định một run, mà chúng ta có thể tô màu dễ dàng

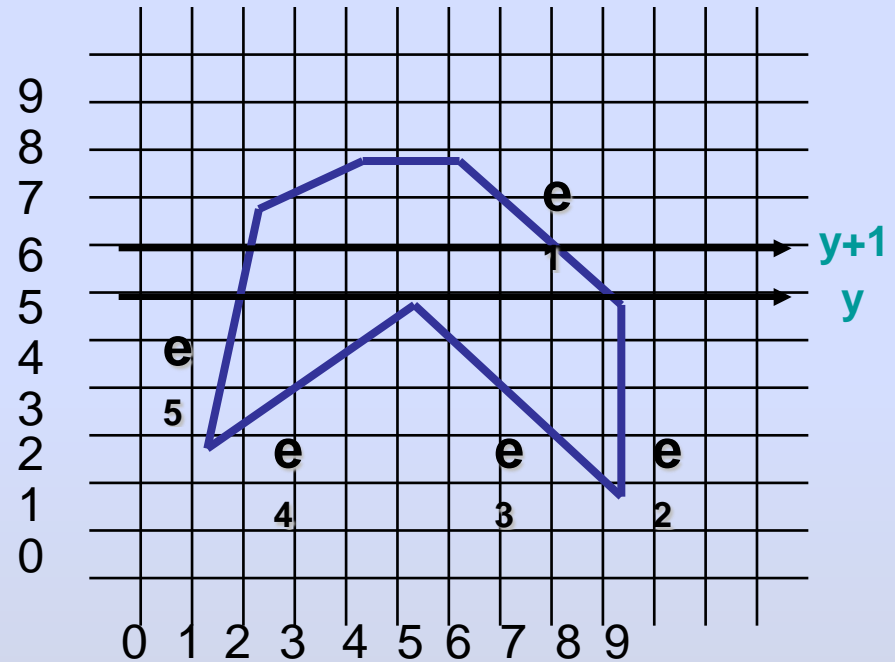
```
tmp = AEL;
while (tmp != NULL)
{
    x1 = tmp.x_int;
    tmp = tmp->next;
    x2 = tmp.x_int;
    tmp = tmp->next;
    for(x = x1; x <= x2; x++)
        putpixel(x,y,color);
}
```



Cập nhật AEL khi tung độ dòng quét thay đổi

Sau khi tô màu tại dòng scanline hiện hành y , AEL phải được cập nhật tại scanline $y+1$:

- Bằng cách so sánh y và y_{upper} của các cạnh trong AEL, ta xác định “dòng scanline mới nằm phía trên cạnh nào đó trong AEL”: xóa cạnh có y vượt quá y_{upper} .
- Giá trị của hoành độ giao điểm thay đổi theo dòng scanline. Khi dòng scanline tăng lên 1 thì x_{int} thay đổi là $1/m$: cập nhật tất cả các cạnh với $x_{int} = x_{int} + recip_slope$



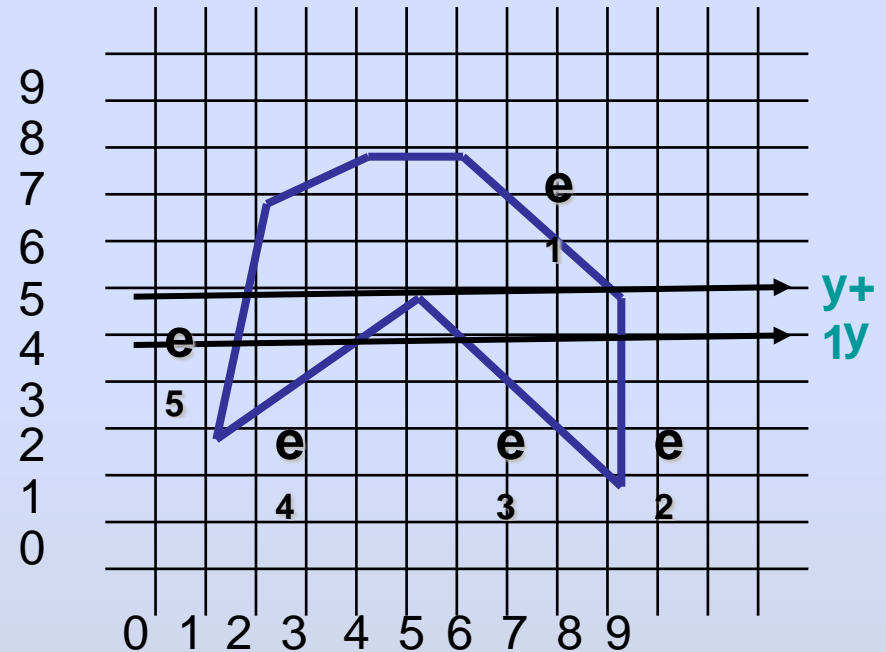
Tại y : $ael = \{e_5, e_4, e_3, e_1\}$

Tại $y+1$: $ael = \{e_5, e_1\}$

Cập nhật AEL khi tung độ dòng quét thay đổi

Sau khi tô màu tại dòng scanline hiện hành y , AEL phải được cập nhật tại scanline $y+1$:

- Khi $y+1$ bằng với y_{lower} của một cạnh thì nó phải được chèn vào AEL (giá trị của cạnh này sẽ trình bày sau trong Edge Table).
- Thứ tự của hoành độ giao điểm có thể đảo ngược khi 2 cạnh giao nhau (đa giác tự cắt): AEL phải được sắp xếp lại.



Tại y : $ael = \{e_5, e_4, e_3, e_2\}$

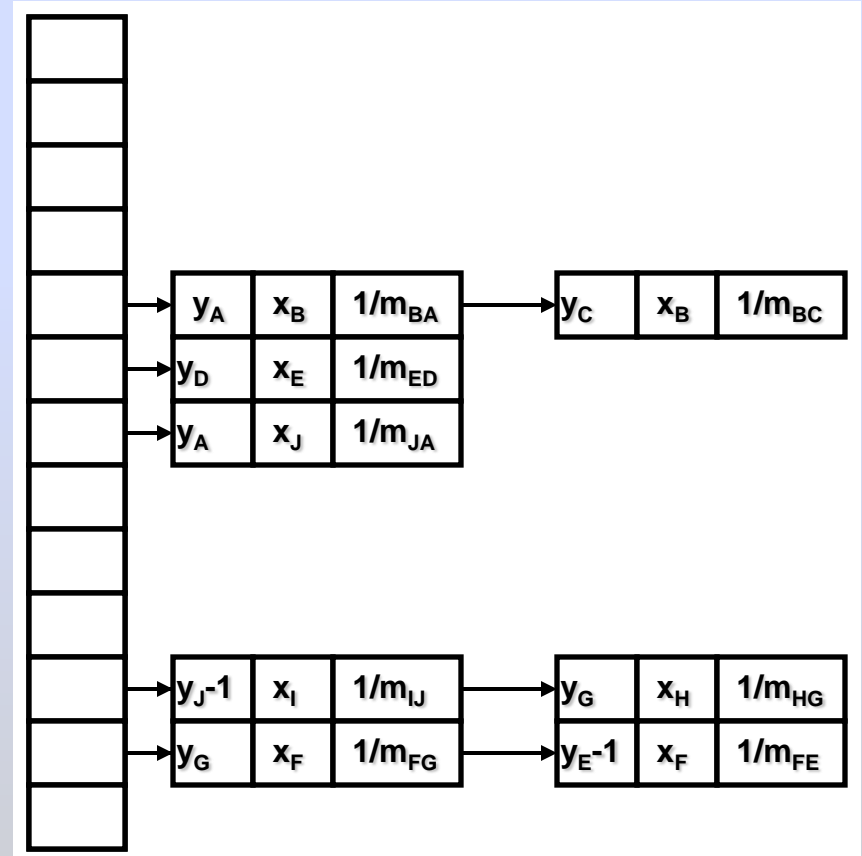
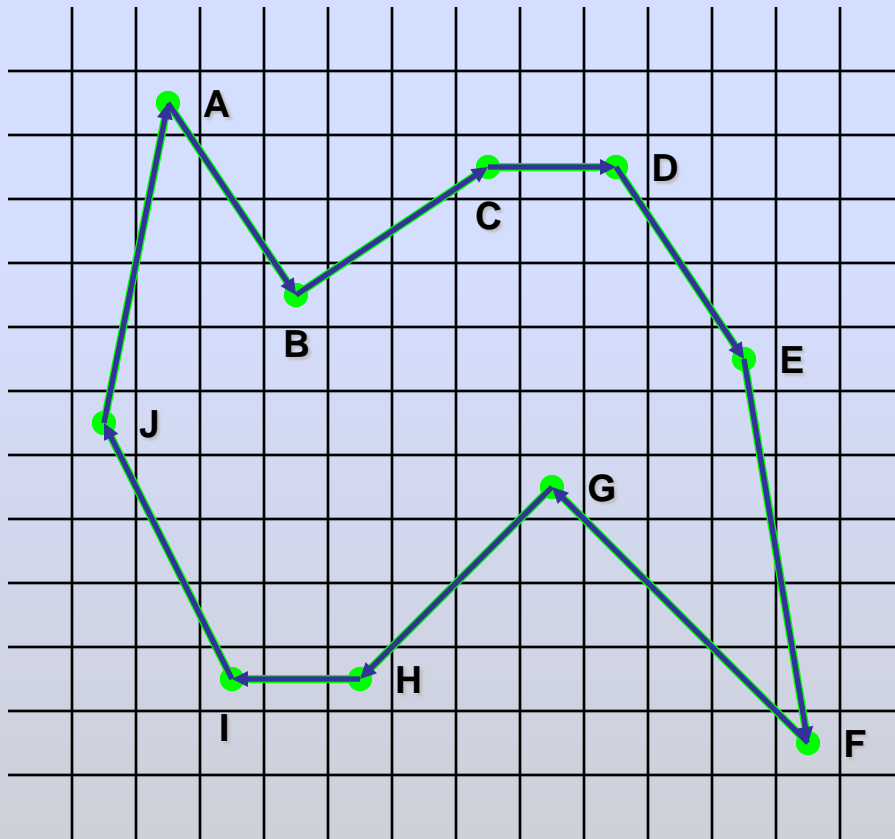
Tại $y+1$: $ael = \{e_5, e_4, e_3, e_1\}$

Sử dụng cấu trúc bảng để quản lý danh sách cạnh

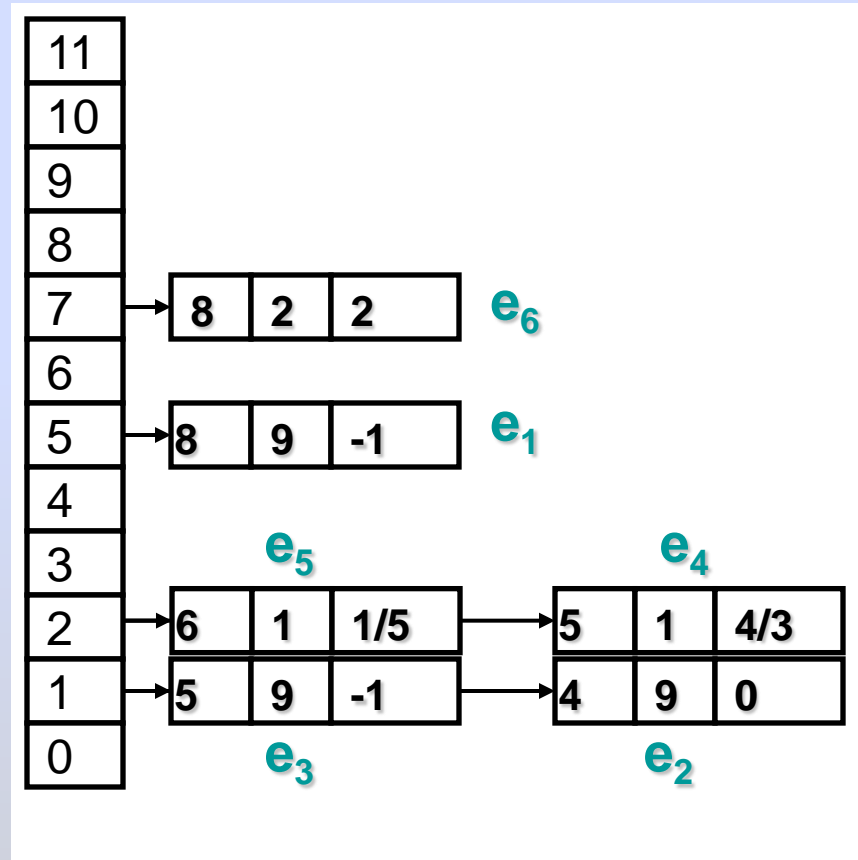
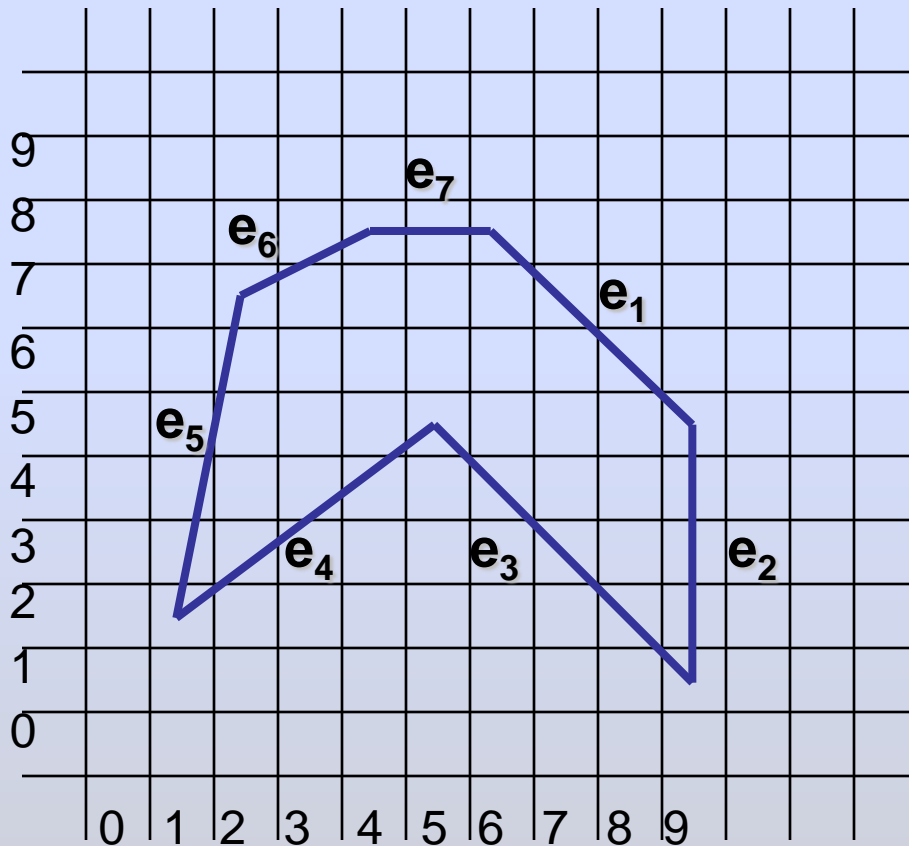
Để xác định cạnh nào được chèn vào AEL, chúng ta phải xét từng đỉnh của đa giác. Tuy nhiên, cấu trúc **EdgeTable** được tạo ra để **lưu trữ thông tin các cạnh** trước khi quá trình tô màu xảy ra, bảo đảm yêu cầu cập nhật nhanh chóng AEL:

- Mỗi cạnh được xác định **y_upper**, **recipe_slope** thông qua đỉnh đa giác, và **x_int** là hoành độ đỉnh dưới của cạnh.
- EdgeTable là một mảng các danh sách các cạnh (như danh sách AEL). EdgeTable[y] chứa danh sách các cạnh có **y_lower = y**

Sử dụng cấu trúc bảng để quản lý danh sách cạnh



Sử dụng cấu trúc bảng để quản lý danh sách cạnh



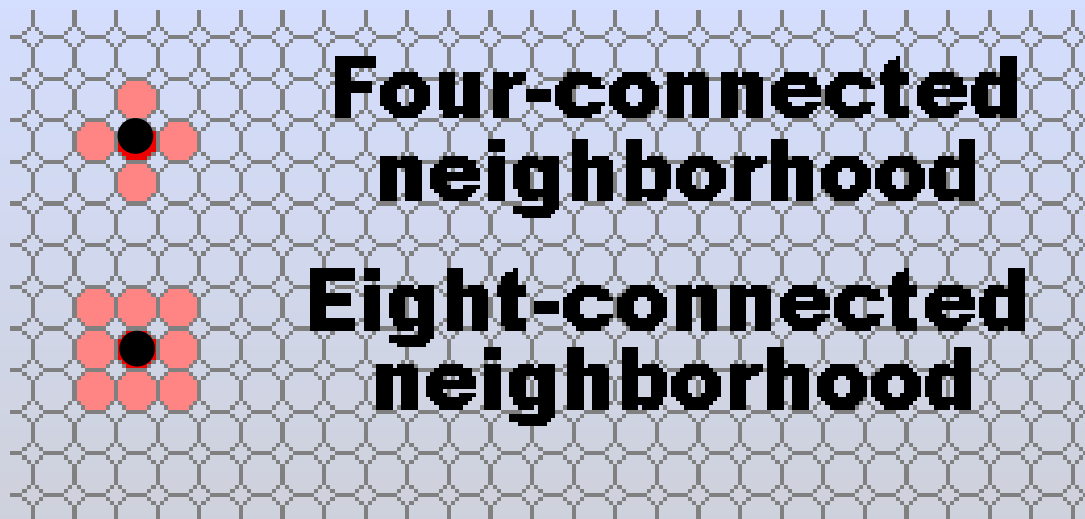
Dùng EdgeTable để cập nhật AEL

Sau khi tạo EdgeTable, AEL dễ dàng được cập nhật thông qua các cạnh có sẵn trong EdgeTable tại dòng scanline y :

- Chèn các cạnh tại EdgeTable[y] vào AEL : nghĩa là dòng scanline bắt đầu cắt các cạnh có $y_lower = y$
- Giá trị ban đầu của x_int là hoành độ của đỉnh dưới nên chính là hoành độ giao điểm ban đầu.

Tô màu theo thuật giải vết dầu loang

- Tô màu cho vùng kín xác định bởi màu của đường biên.
- Dùng thuật giải đệ quy xét màu pixel, và các pixel lân cận.
- Sử dụng lân cận 4, hoặc lân cận 8.



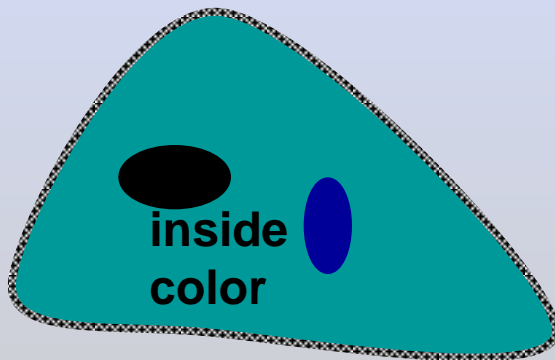
Tô màu theo thuật giải vết dầu loang

- Interior defined

- Tất cả các pixel trong vùng có cùng một màu, gọi là **inside-color**
- Các pixel trên biên không có màu này
- Có thể có lỗ trong vùng

- Boundary defined

- Các pixel thuộc biên có cùng màu – **boundary-color**
- Các pixel trong vùng không có màu này
- Nếu một số pixel trong vùng có màu boundary-color thì vùng sẽ chứa lỗ



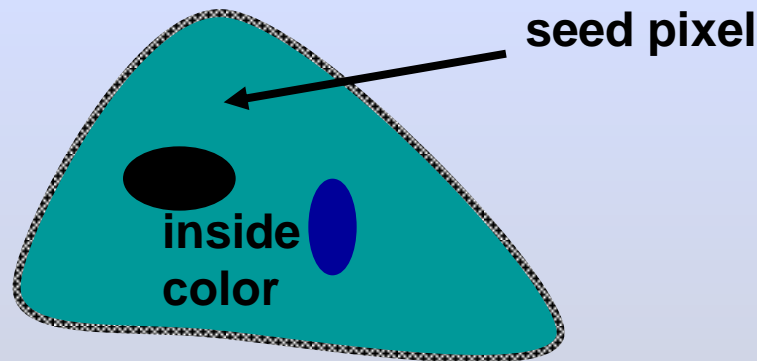
Interior-defined



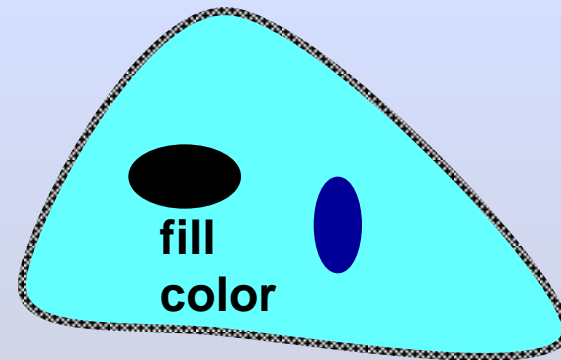
Boundary-defined

Tô màu theo thuật giải vết dầu loang

- Đổi màu của tất cả các interior-pixel thành màu tô.
- Quá trình tô màu bắt đầu từ một điểm (**seed pixel**) thuộc phía trong vùng tô và lan truyền khắp vùng tô.



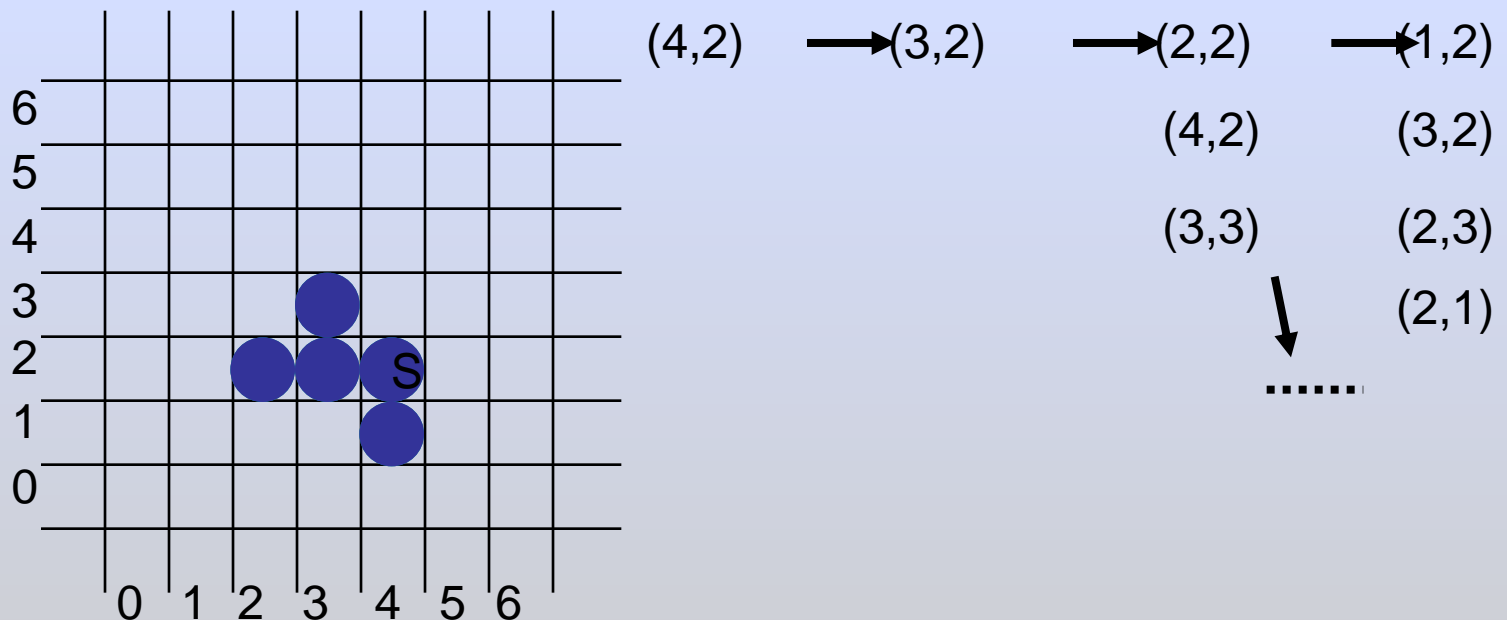
Interior-defined



Recursive Flood-Fill

Tô màu theo thuật giải vết dầu loang

- Nếu pixel tại (x,y) thuộc vùng trong – màu của pixel đó là **inside-color** thì
 - Đổi màu của nó thành **fill-color**
 - Áp dụng quá trình trên cho 4 điểm lân cận nó (4-connected).
- Ngược lại, không làm gì.



Tô màu theo thuật giải vết dầu loang

```
void FloodFill(int x, int y, int inside_color, int
    fill_color)
{
    if (getpixel(x,y) == inside_color)
    {
        putpixel(x,y,fill_color);
        FloodFill(x-1,y, inside_color, fill_color);
        FloodFill(x+1,y, inside_color, fill_color);
        FloodFill(x,y+1, inside_color, fill_color);
        FloodFill(x,y-1, inside_color, fill_color);
    }
}
```

Bài tập

1. Trình bày thuật giải scan line theo đường quét nghiêng 45 độ (0.5đ).
2. Thực hành: cài đặt hai thuật giải scan line và flood-fill.

Hỏi đáp

