# MIỀN HAI MÀU

Cho một bảng kích thước được chia làm lưới ô vuông đơn vị. Các hàng của bảng được đánh số từ 1 tới từ trên xuống và các cột của bảng được đánh số từ 1 tới từ trái qua phải. Ô nằm trên giao của hàng và cột được gọi là ô và ô đó cần phải tô bằng một màu có mã màu là một số nguyên , hai màu khác nhau có mã khác nhau.

Một tập các ô của bảng được gọi là một miền nếu giữa hai ô bất kỳ của miền đó, ta có thể đi từ ô này sang ô kia bằng các phép di chuyển qua các ô kề cạnh.

**Yêu cầu:** Tìm một miền gồm nhiều ô nhất sao cho có thể tô màu các ô của miền bằng không quá hai màu.

**Dữ liệu:** Vào từ file văn bản COLORING.INP

* Dòng 1 chứa hai số nguyên dương
* dòng tiếp theo, dòng thứ chứa số nguyên dương, số thứ là

**Kết quả:** Ghi ra file văn bản COLORING.OUT một số nguyên duy nhất là số ô của miền tìm được.

**Ví dụ**

|  |  |
| --- | --- |
| COLORING.INP | COLORING.OUT |
| 5 6  1 1 1 1 3 1  2 3 4 5 4 1  1 2 1 2 5 1  6 7 8 1 6 2  1 1 1 1 7 2 | 14 |

# Cách giải:

Liệt kê các miền liên thông gồm các ô cần tô cùng màu (gọi tắt là miền đơn sắc) (Đặt $M rồi cài DFS cho nhanh), đánh số các miền đơn sắc này từ 1 trở đi.

* Gọi là số hiệu miền đơn sắc chứa ô
* Gọi là màu của miền đơn sắc thứ , tức là màu của tất cả các ô mà
* Gọi là số ô (diện tích) của miền đơn sắc thứ , tức là số ô mà .

Xét tất cả các “khe” giữa hai ô kề cạnh, tức là với mỗi ô xét ô bên phải nó và bên dưới nó nếu có (tổng cộng khe như vậy). Với mỗi khe giữa hai ô khác màu (), tạo một bản ghi chứa 4 thông tin sau:

* Miền đơn sắc chứa ô :
* Màu của ô :
* Miền đơn sắc chứa ô :
* Màu của ô :

(Thực ra không cần lưu và , chỉ cần và là đủ vì và )

Lưu tất cả các bản ghi này lại, mỗi bản ghi ứng với một khe nối giữa hai miền đơn sắc (màu ) và (màu ). Không giảm tính tổng quát, giả sử (Nếu lớn hơn thì ta đảo và cũng như đảo ).

**Kỹ thuật giảm độ phức tạp ở đây dựa vào nhận xét rằng số khe triệu. Và để tìm miền liên thông lớn nhất gồm 2 màu XANH và ĐỎ thì ta chỉ cần xét các khe có XANH và ĐỎ mà thôi.**

Sort lại danh sách các khe để các khe có cùng giá trị và dồn lại một đoạn liên tiếp trong dãy (đó là lý do ta quy ước ). Xây dựng cấu trúc dữ liệu Disjoint-set forest (DSF) biểu diễn các tập miền đơn sắc, mỗi tập trong CTDL này đi kèm với giá trị là tổng diện tích các miền đơn sắc tập.

* MakeSet(): Tạo tập chỉ chứa 1 miền đơn sắc :
  + ;
  + ;
* FindSet(): Tìm tập chứa , như bình thường (đi từ lên gốc theo , nén đường)
* Union(): Hợp hai tập . Chú ý khi hợp xong thì của tập mới bằng tổng hai tập cũ. Giá trị chỉ có nghĩa khi là một gốc cây biểu diễn tập hợp trong DSF.
* Khởi tạo: Duyệt từng khe trong đoạn, với mỗi khe nối hai miền đơn sắc , gọi và , cập nhật MaxArea bằng cái lớn nhất vừa đặt.
* Xử lý: Duyệt từng khe trong đoạn, với mỗi khe mà thì . Cập nhật MaxArea theo của tập hợp thành nếu nó lớn hơn MaxArea cũ.
* Kết thúc việc xử lý đoạn liên tiếp **MaxArea cho ta diện tích miền lớn nhất chỉ gồm hai màu và .**

Còn lại thì xong rồi. Xử lý mỗi đoạn liên tiếp cho ta diện tích miền lớn nhất gồm 2 màu ứng với hai giá trị của các khe trong đoạn đó.

Độ phức tạp: Ngoài các thao tác tìm miền đơn sắc và sắp xếp khe. Việc xử lý một đoạn liên tiếp độ dài mất thời gian . Tổng độ dài các đoạn liên tiếp số khe triệu).