

Giao hữu bóng đá

Có n đội bóng đá (các đội được đánh số từ 1 đến n) thi đấu giao hữu theo thể thức vòng tròn, mỗi đội sẽ thi đấu với tất cả các đội còn lại. Để tăng sự hấp dẫn, Ban tổ chức quyết định mỗi trận đấu sẽ luôn được phân định thắng thua, cụ thể nếu trong thời gian thi đấu mà hai đội không phân thắng thua thì sẽ được phân định bằng đá luân lưu. Sau khi kết thúc giải đấu, với cặp số giới hạn L, R ($1 \leq L < R \leq n$), Ban tổ chức muốn thống kê một trong hai thông tin thú vị như sau:

- Chuỗi thắng dài nhất: Cần chọn một danh sách nhiều nhất các đội phân biệt p_1, p_2, \dots, p_r mà $L \leq p_1, p_2, \dots, p_r \leq R$ và đội trước thắng đội liền sau trong danh sách.
- Vòng thắng nhỏ nhất: Cần chọn một danh sách ít nhất các đội phân biệt q_1, q_2, \dots, q_s mà $L \leq q_1, q_2, \dots, q_s \leq R$ và đội trước thắng đội liền sau trong danh sách, ngoài ra đội cuối cùng trong danh sách lại thắng đội đầu tiên trong danh sách.

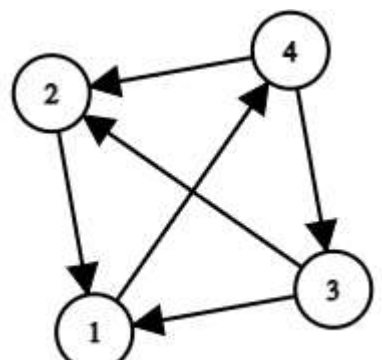
Yêu cầu: Cho biết kết quả tất cả các trận đấu của n đội biểu diễn bằng mảng c , trong đó $c_{ij} = 0$ nghĩa là đội i thua đội j ($i \neq j$), trái lại $c_{ij} = 1$ cho biết đội i thắng đội j ($c_{ji} = 1 - c_{ij}$; $c_{ii} = 0$), T cặp số giới hạn $L_1, R_1, L_2, R_2, \dots, L_T, R_T$ và số nguyên k , với mỗi cặp số giới hạn nếu $k = 1$ đưa ra thông tin về chuỗi thắng dài nhất, nếu $k = 2$ đưa ra thông tin về vòng thắng nhỏ nhất thỏa mãn.

Dữ liệu:

- Dòng đầu tiên ghi hai số nguyên dương n, T, k ($n \leq 1000; T \leq 100; 1 \leq k \leq 2$);
- Dòng thứ hai chứa T cặp số $L_1, R_1, L_2, R_2, \dots, L_T, R_T$ mô tả T cặp số giới hạn;
- Tiếp theo là n dòng, trong đó số thứ j ($1 \leq j \leq n$) của dòng thứ i ($1 \leq i \leq n$) chứa số nguyên c_{ij} .

Kết quả: Ghi ra T dòng, mỗi dòng tương ứng câu trả lời cho một cặp giới hạn có khuôn dạng: Số đầu tiên ghi -1 nếu không tìm được dãy các đội thỏa mãn, ngược lại ghi số lượng đội trong danh sách tìm được; Nếu có dãy thỏa mãn, tiếp theo ghi chỉ số các đội lần lượt trong danh sách.

Ví dụ:

Input	Output	Hình minh họa
4 1 1 1 4 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0	4 2 1 4 3	
4 2 2 1 4 1 3 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0	3 1 4 2 -1	

Ràng buộc:

- Có 10% số test của bài có $n = 3; T \leq 3$;
- Có 20% số test khác của bài có $n \leq 20; T \leq 3$ và $k = 1$;
- Có 20% số test khác của bài có $n \leq 20; T \leq 3$ và $k = 2$;
- Có 25% số test khác của bài có $k = 1$;
- Có 25% số test còn lại của bài có $k = 2$.

Liên thông

Một khu du lịch do Alice quản lí là một quần đảo được mô tả bằng một bảng kí tự gồm m hàng và n cột, các hàng được đánh số từ 1 đến m từ trên xuống dưới, các cột được đánh số từ 1 đến n từ trái sang phải. Ô nằm giao giữa hàng i và cột j gọi là ô (i, j) và chứa kí tự '1' hoặc '0', trong đó '1' mô tả vùng đất (gọi là ô đất), '0' mô tả vùng nước (gọi là ô nước). Một đảo là một vùng liên thông gồm các ô đất mà hai ô có thể đi được đến nhau thông qua các ô đất chung cạnh. Khách du lịch có thể chọn xuất phát tại một ô đất và đi bộ thăm hết các ô đất trên đảo (chứa ô đất xuất phát) bằng cách di chuyển thông qua các ô chung cạnh.

Với mong muốn khách du lịch có thể đi bộ và thăm quan được nhiều hơn, Alice lên kế hoạch dựng cầu treo trên các ô nước để liên thông một số đảo. Cụ thể, sau khi dựng cầu treo ở các ô nước, du khách có thể di chuyển vào các ô nước đó như đi vào các ô đất, Alice mong muốn tồn tại cách thăm quan cho du khách mà tổng số ô đất du khách có thể đi thăm quan được lớn hơn hoặc bằng s và số lượng ô nước cần dựng cầu treo là ít nhất.

Yêu cầu: Cho bảng kí tự mô tả quần đảo và T phương án tương ứng với T số nguyên dương s_1, s_2, \dots, s_T , với mỗi phương án s_k ($1 \leq k \leq T$) hãy giúp Alice tính số ô nước ít nhất cần dựng cầu treo để tổng số ô đất mà khách du lịch có thể đi bộ và thăm được lớn hơn hoặc bằng s_k .

Dữ liệu:

- Dòng đầu gồm ba số nguyên dương m, n, T ($T \leq 100$);
- Dòng thứ i ($1 \leq i \leq m$) trong số m dòng tiếp theo chứa một xâu độ dài n chỉ gồm hai loại kí tự '1' và '0';
- Dòng cuối cùng chứa T số nguyên dương s_1, s_2, \dots, s_T ($s_k \leq m \times n$).

Kết quả:

- Gồm T số nguyên tương ứng với T phương án, mỗi số là số ô nước ít nhất cần dựng cầu treo hoặc đưa ra -1 nếu không tồn tại cách dựng thỏa mãn.

Ràng buộc:

Gọi c là số đảo trong khu du lịch, trong tất cả các test đều thỏa mãn $1 \leq c \leq 10$ và $m, n \leq 100$, ngoài ra:

- Có 10% số test ứng với 10% số điểm của bài có $c = 1; m, n \leq 30$;
- Có 25% số test ứng khác với 25% số điểm của bài có $c \leq 2; m, n \leq 30$;
- Có 25% số test ứng khác với 25% số điểm của bài có $c \leq 3; m, n \leq 30$;
- Có 20% số test ứng khác với 20% số điểm của bài $c \leq 8; m, n \leq 30$;
- Có 20% số test còn lại ứng với 20% số điểm của bài không có ràng buộc nào thêm.

Ví dụ:

Input	Output	Input	Output
5 6 3 000001 011101 010011 000001 001100 9 10 15	1 2 -1	3 5 6 10101 01010 10101 1 2 3 4 5 6	0 1 1 1 2 2

cis

Alice và Bob cùng được cho một đơn đồ thị G gồm n đỉnh. Alice đã chọn cho mình tập đỉnh C đầy đủ, Bob cũng chọn cho mình tập độc lập I . Hai bạn được trao đổi thông tin để trả lời hai tập C và I có đỉnh chung hay không?

Alice và Bob chỉ được truyền thông tin lần lượt cho nhau, Alice thực hiện việc truyền thông tin trước, mỗi lượt hai bạn được gửi theo cách sau: Chỉ gửi đúng một số nguyên v có giá trị $[-n, n]$, trong đó, nếu $v > 0$ được tính là một lần gửi thực sự, $v = 0$ được coi là không tính lần gửi, $v < 0$ được coi là không tính lần gửi và trả lời đỉnh $|v|$ là đỉnh chung và sẽ kết thúc việc truyền tin. Nếu cả hai bạn đều gửi thông tin bằng 0 cho nhau hệ thống sẽ hiểu là hai tập C, I không có đỉnh chung.

#include "cis.h" và viết các hàm sau để truyền thông tin như mô tả ở trên.

```
#include <vector>

namespace personA {
    void initA(int _N, std::vector<pair<int, int>> _E, std::vector<int> _C);
    int A2B(int b);
    // a = 0 -> ko truyền; a < 0 -> trả lời đỉnh |a|; a > 0 -> truyền a;
}

namespace personB {
    void initB(int _N, std::vector<pair<int, int>> _E, std::vector<int> _I);
    int B2A(int a);
}
```