

## 1. ĐƯỜNG MỘT CHIỀU

Một hệ thống giao thông có  $n$  địa điểm đánh số từ 1 tới  $n$  và  $m$  tuyến đường đánh số từ 1 tới  $m$ . Tuyến đường thứ  $i$  nối từ địa điểm  $u_i$  tới địa điểm  $v_i$ , cho phép đi từ  $u_i$  tới  $v_i$  theo **một chiều**.

Ta nói địa điểm  $s$  có thể đi tới địa điểm  $t$  nếu tồn tại dãy  $s = p_0, p_1, \dots, p_k = t$  sao cho có tuyến đường cho phép đi từ  $p_{i-1}$  tới  $p_i$  ( $\forall i = 1, 2, \dots, k$ )

**Yêu cầu:** Hãy xây dựng thêm một số ít nhất các tuyến đường một chiều để hệ thống giao thông đảm bảo được sự đi lại giữa hai địa điểm bất kỳ.

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

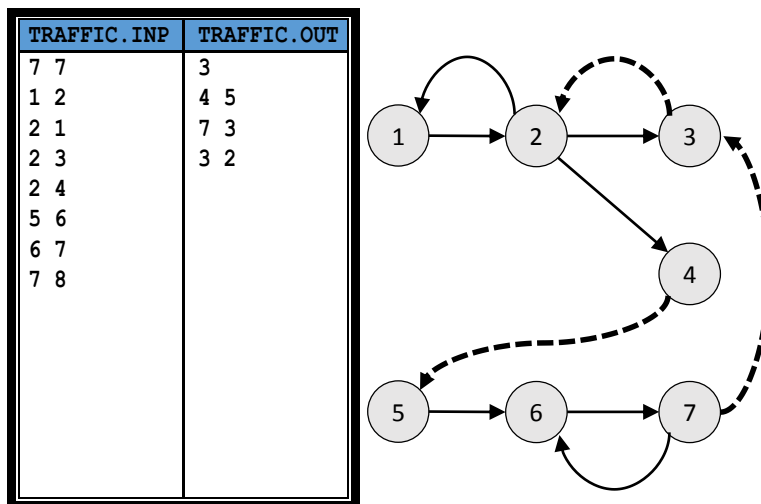
- Dòng 1: Chứa hai số nguyên dương  $n \leq 10^5$ ;  $m \leq 2 \cdot 10^5$
- $m$  dòng tiếp theo, dòng thứ  $i$  chứa hai số nguyên dương  $u_i, v_i$

**Kết quả:** Ghi ra file văn bản TRAFFIC.OUT

- Dòng 1: Ghi số  $k$  là số tuyến đường cần xây dựng thêm
- $k$  dòng tiếp theo, mỗi dòng ghi hai số  $x, y$  tương ứng với một tuyến đường cần xây dựng thêm để đi theo một chiều từ  $x$  tới  $y$

*Các số trên một dòng của input/output files được/phải ghi cách nhau ít nhất một dấu cách*

**Ví dụ**



## 2. BIẾN ĐỔI BẢNG SỐ

Cho hai bảng ô vuông  $A, B$  kích thước  $m \times n$ , mỗi ô của mỗi bảng chứa đúng một chữ số nhị phân: 0 hoặc 1.

Đánh số các hàng của hai bảng từ 1 tới  $m$  theo thứ tự từ trên xuống dưới và các cột của hai bảng từ 1 tới  $n$  theo thứ tự từ trái qua phải. Với một số nguyên  $i$ , xét phép đảo bit tham số  $i$ :  $Invert(i)$ :

- Nếu  $1 \leq i \leq m$ , phép đảo bit  $Invert(i)$  sẽ xét hàng  $i$  của bảng  $A$ , thay tất cả các chữ số 0 thành 1 và 1 thành 0
- Nếu  $-n \leq i \leq -1$ , phép đảo bit  $Invert(i)$  sẽ xét cột  $-i$  của bảng  $A$ , thay tất cả các chữ số 0 thành 1 và 1 thành 0

Nếu tham số  $i$  không thỏa mãn một trong hai điều kiện trên, phép đảo bit  $Invert(i)$  bị coi là không hợp lệ.

*Yêu cầu: Hãy chỉ ra một số ít nhất các phép đảo bit hợp lệ để biến bảng  $A$  thành bảng  $B$  hoặc cho biết rằng không thể biến đổi bảng  $A$  thành bảng  $B$  bằng những phép đảo bit.*

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

- Dòng 1 chứa hai số nguyên dương  $m, n \leq 500$  cách nhau ít nhất một dấu cách
- $m$  dòng tiếp theo, dòng thứ  $i$  chứa  $n$  số nhị phân liên nhau, chữ số thứ  $j$  là số ghi trên ô  $(i, j)$  của bảng  $A$
- $m$  dòng tiếp theo, dòng thứ  $i$  chứa  $n$  số nhị phân liên nhau, chữ số thứ  $j$  là số ghi trên ô  $(i, j)$  của bảng  $B$

**Kết quả:** Ghi ra file văn bản INVERSE.OUT

- Dòng 1: Ghi số phép đảo bit cần thực hiện ( $k$ ), nếu không có phương án thực hiện, dòng này ghi số -1.
- Nếu có phương án thực hiện, dòng 2 ghi  $k$  tham số của các phép đảo bit cần thực hiện cách nhau đúng một dấu cách.

**Ví dụ:**

INVERSE.INP	INVERSE.OUT
5 5	2
10110	2 -3
01001	
10111	
11010	
00000	
10010	
10010	
10011	
11110	
00100	

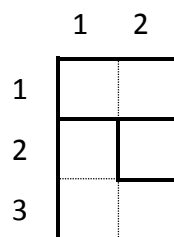
### 3. PHÁ TƯỜNG

Có một toà lâu đài hình chữ nhật kích thước  $m \times n$  ô vuông đơn vị. Các dòng ô vuông được đánh số từ 1 tới  $m$  từ trên xuống dưới và trên mỗi dòng, các ô được đánh số theo thứ tự từ 1 tới  $n$  từ trái qua phải. Trong lâu đài có một số bức tường giữa các ô, định nghĩa một đường đi giữa hai ô là một cách di chuyển từ ô này tới ô kia qua các ô kề cạnh mà hai ô kề cạnh trong mỗi bước di chuyển không bị tường chắn ở giữa hai ô đó. Các bức tường phân lâu đài ra làm nhiều phòng, mỗi phòng gồm một số ô. Cách chia phòng thoả mãn hai điều kiện:

- Tồn tại đường đi giữa hai ô bất kỳ trong cùng một phòng
- Không tồn tại đường đi giữa hai ô thuộc hai phòng khác nhau

Giả thiết rằng quanh lâu đài có tường bao bọc.

Ví dụ dưới đây là một lâu đài  $3 \times 2$  có 3 phòng.



Hãy cho biết lâu đài có bao nhiêu phòng, cho biết diện tích của phòng rộng nhất và tìm cách phá đi đúng một bức tường để được một phòng rộng nhất có thể

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

- Dòng 1: Ghi hai số  $m, n$  ( $1 \leq m, n \leq 1000$ )
- Các dòng tiếp theo, mỗi dòng ghi 4 số nguyên  $x_1, y_1, x_2, y_2$  cho ta thông tin: có tường ngăn ô  $(x_1, y_1)$  và ô  $(x_2, y_2)$ . Có ít nhất một bức tường ngăn các ô trong lâu đài.

**Kết quả:** Ghi ra file văn bản DWALL.OUT

- Dòng 1: Ghi số phòng của lâu đài
- Dòng 2: Ghi diện tích của phòng rộng nhất
- Dòng 3: Ghi bốn số nguyên  $u_1, v_1, u_2, v_2$  cho biết cần phá tường ngăn ô  $(u_1, v_1)$  và ô  $(u_2, v_2)$ . Bắt buộc phải phá đúng một bức tường.
- Dòng 4: Ghi diện tích của phòng rộng nhất thu được sau khi phá tường

*Các số trên một dòng của Input/Output files được/phải ghi cách nhau ít nhất một dấu cách*

Ví dụ:

DWALL . INP	DWALL . OUT
3 2	3
1 1 2 1	3
1 2 2 2	1 1 2 1
2 1 2 2	5
2 2 3 2	