

## BÀI LẬP TRÌNH

### Bài 1 GRAPH.CPP

Cho một đồ thị vô hướng gồm  $n$  đỉnh đánh số từ 1 tới  $n$  và  $m$  cạnh đánh số từ 1 tới  $m$ . Cạnh thứ  $i$  nối giữa hai đỉnh  $u_i, v_i$ . Nếu ta xoá đi một đỉnh nào đó của đồ thị, số thành phần liên thông của đồ thị có thể tăng lên. Nhiệm vụ của bạn là với mỗi đỉnh, hãy tính xem nếu ta xoá đỉnh đó đi thì đồ thị mới nhận được có bao nhiêu thành phần liên thông.

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

- Dòng đầu chứa hai số nguyên dương  $n, m$  ( $n \leq 20000; m \leq 50000$ )
- $m$  dòng sau, dòng thứ  $i$  chứa hai số nguyên dương  $u_i, v_i$ .

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

- $n$  dòng, dòng thứ  $j$  cho biết số thành phần liên thông của đồ thị nếu ta xoá đi đỉnh  $j$ .

**Ví dụ**

GRAPH.INP	GRAPH.OUT
4 3	1
1 2	3
2 3	1
2 4	1

Chú ý: Ít nhất 60% số điểm ứng với các test có  $n \leq 1000; m \leq 2000$

#### \* Hướng dẫn thuật toán

- Đây là bài toán điển hình về tìm khớp trên đồ thị. Nếu đỉnh  $u$  không phải là khớp thì số lượng thành phần liên thông không thay đổi, nếu đỉnh  $u$  là khớp thì số lượng thành phần liên thông sẽ được tăng lên.
- Vấn đề ở đây là làm thế nào để được số lượng thành phần liên thông sau khi đã loại bỏ một khớp  $u$  ra khỏi đồ thị. Có thể giải quyết vấn đề này như sau: trong quá trình DFS sử dụng thêm một mảng để lưu số lượng đỉnh con của đỉnh  $u$  là slcon[u] (Chú ý trường hợp

có cung ngược ). Khi đó nếu số lượng thành phần liên thông ban đầu của đồ thị là k thì tiếp theo sẽ có ba khả năng như sau:

- + Khả năng 1: u là khớp nhưng không phải đỉnh gốc của DFS thì số lượng thành phần liên thông sau khi xóa đỉnh khớp u là:  $k + \text{slcon}[u]$ .
- + Khả năng 2: u là khớp nhưng lại là đỉnh gốc của DFS thì số lượng thành phần liên thông sau khi xóa đỉnh khớp u là:  $k + \text{slcon}[u] - 1$ .
- + Khả năng 3: u là đỉnh đơn lẻ thì số lượng thành phần liên thông là :  $k - 1$ .

## Bài 2:

### Đường 1 chiều

**Tên file: ONEWAY.cpp**

Một hệ thống giao thông gồm có N nút giao thông đánh số từ 1 đến N và M đường hai chiều nối một số cặp nút, không có hai đường nối cùng một cặp nút. Hệ thống đảm bảo đi lại giữa hai hòn bất kỳ. Để đảm bảo an toàn, người ta quyết định rằng các đường hai chiều trước đây nay sẽ thành một chiều, và vấn đề ở chỗ chọn chiều cho mỗi đường như thế nào.

Hãy tìm cách định hướng các cạnh sao cho hệ thống vẫn đảm bảo đi lại giữa hai cặp nút bất kỳ.

**INPUT: ONEWAY.INP**

- Dòng đầu ghi hai số nguyên dương N, M ( $1 < N < 50000$ ,  $1 < M < 100000$ ).
- M dòng tiếp theo, mỗi dòng thể hiện một đường hai chiều gồm u, v là chỉ số hai nút mà nó nối tới .

**OUTPUT: ONEWAY.OUT**

- Dòng đầu ghi 1/0 tương ứng với có tìm được phương án thỏa mãn hay không .
- Nếu có, M dòng tiếp theo mỗi dòng thể hiện sự định hướng một cạnh bao gồm hai số u, v với ý nghĩa định hướng cạnh (u,v) thành đường một chiều từ u đến v.

Ví dụ

Test 1		Test 2	
ONEWAY.INP	ONEWAY.OUT	ONEWAY.INP	ONEWAY.OUT
4 5	1	4 4	0
1 2	1 2	1 2	
2 3	2 3	2 3	
2 4	2 4	3 4	
3 4	3 4	3 1	

### Bài 3. Bảo vệ

Một thành phố có N địa điểm chiến lược và M con đường một chiều giữa các địa điểm. Là thị trưởng của thành phố, bạn sẽ phải bảo vệ an toàn cho N địa điểm này.

Để có thể bảo vệ cho các địa điểm, bạn phải xây dựng các đòn cảnh sát tại một vài địa điểm. Đòn cảnh sát tại địa điểm i có thể bảo vệ cho địa điểm j nếu  $i = j$  hoặc cảnh sát có thể đi tuần tới địa điểm j từ i và có thể quay trở lại đòn tại địa điểm i.

Để có thể xây dựng được các đòn cảnh sát cần phải mất chi phí, do địa hình mỗi địa điểm là khác nhau nên chi phí xây dựng đòn cũng có thể khác nhau.

Bạn phải xác định số tiền nhỏ nhất để xây dựng các đòn cảnh sát để có thể bảo vệ được tất cả N địa điểm, hơn nữa bạn phải đưa ra có bao nhiêu cách xây dựng để đảm bảo chi phí nhỏ nhất đó.

INPUT: SECURITY.INP

- Dòng 1 chứa số nguyên dương N ( $1 \leq N \leq 10^5$ )
- Dòng 2 chứa N số nguyên, trong đó số nguyên thứ i là chi phí để xây dựng đòn cảnh sát tại địa điểm i (chi phí  $\leq 10^9$ ).
- Dòng 3 chứa số nguyên M ( $0 \leq M \leq 3 * 10^5$ )
- M dòng tiếp theo, mỗi dòng chứa hai số nguyên dương u và v ( $1 \leq u, v \leq n; u \neq v$ ) biểu diễn một con đường một chiều nối từ địa điểm u tới v. Không có nhiều hơn 1 con đường nối giữa 2 địa điểm.

OUTPUT: SECURITY.OUT

- Một dòng duy nhất chứa hai số, số thứ nhất là chi phí nhỏ nhất để xây dựng các đòn cảnh sát, số thứ hai là số phương án xây dựng ( $\text{mod } (10^9 + 7)$ ).

Ví dụ:

SECURITY.INP	SECURITY.OUT
5 2 8 0 6 0	8 2

6	
1 4	
1 3	
2 4	
3 4	
4 5	
5 1	

#### \* Hướng dẫn thuật toán:

- Sử dụng thuật toán Tajan để tìm các thành phần liên thông mạnh, tại mỗi thành phần liên thông mạnh sẽ xây dựng một đòn cảnh sát có chi phí xây dựng là nhỏ nhất, và đếm số lượng các đòn có cùng chi phí nhỏ nhất đó.
- Tổng số tiền xây dựng là tổng số tiền xây dựng các đòn có chi phí nhỏ nhất ở mỗi thành phần liên thông.
- Số cách xây dựng là tích của số lượng các đòn cùng chi phí nhỏ nhất ở mỗi thành phần liên thông.

#### Bài 4. Đường đua dài nhất

Mạng giao thông của thành phố ByteLand có N nút giao thông. Giữa hai nút giao thông có tối đa một đường phố hai chiều nối trực tiếp giữa chúng. Nhân dịp chào mừng kỷ niệm 100 năm ngày thành lập, Lãnh đạo thành phố quyết định tổ chức một cuộc đua xe đạp. Đường đua xe đạp sẽ xuất phát từ một nút bất kỳ, qua một số nút khác và trở lại nút ban đầu sao cho không có nút nào (trừ nút xuất phát) đường đua qua đó hai lần. Thật ngạc nhiên, mạng lưới giao thông của thành phố cho phép lập nhiều đường đua xe đạp như vậy tuy nhiên: **mỗi một đường phố sẽ thuộc không quá một đường đua thỏa mãn điều kiện nêu trên.**

Hãy tìm đường đua có số đường phố khác nhau đi qua nhiều nhất.

INPUT: MAXCYCLE.INP

- Dòng đầu ghi hai số nguyên N, M là số nút giao thông và số đường phố trong thành phố ( $N \leq 5.000$ ,  $M \leq 100000$ )

- M dòng tiếp theo, mỗi dòng ghi hai số nguyên u, v thể hiện hai nút của một đường phố

OUTPUT: MAXCYCLE.OUT

- Ghi một số nguyên duy nhất là độ dài (số lượng đường phố khác nhau) của đường đua dài nhất.

Ví dụ:

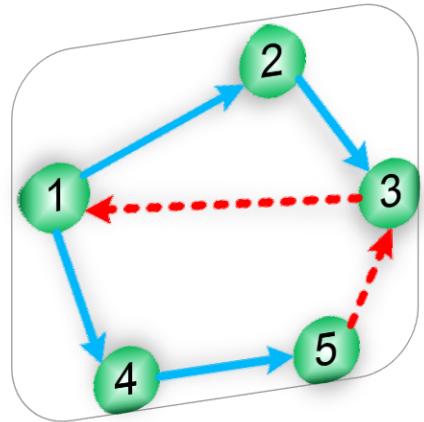
<b>MAXCYCLE.INP</b>	<b>MAXCYCLE.OUT</b>
<b>7 8</b>	
<b>3 4</b>	
<b>1 4</b>	
<b>1 3</b>	
<b>7 1</b>	
<b>2 7</b>	
<b>7 5</b>	
<b>5 6</b>	
<b>6 2</b>	

#### \* Hướng dẫn thuật toán

- Đây là một bài toán tìm thành phần song liên thông với số lượng đỉnh lớn nhất và  $\geq 3$ .

## Bài 5. Tàu cao tốc.

Có  $n$  điểm tập trung dân cư đông đúc. Các điểm này được đánh số từ 1 đến  $n$  ( $1 \leq n \leq 10^4$ ). Mạng lưới giao thông công cộng là  $m$  đường xe lửa cao tốc một ray, mỗi đường nối một cặp điểm dân cư và chạy hai chiều ( $0 \leq m \leq 10^5$ ), và mọi cặp điểm đều có thể đi đến được với nhau. Để tránh sự va chạm giữa các con tàu cao tốc khi chúng có thể đi ngược chiều trên cùng một đường, chính quyền thành phố quyết định sửa lại các con đường đó thành một chiều. Tuy nhiên, sau khi thay đổi thì lại có một vấn đề bất cập xảy ra, đó là: tồn tại các cặp điểm tập trung dân cư không thể đi đến được nhau.



Chính vì vậy, chính quyền lại thêm một quyết định nữa, đó là sẽ xây dựng thêm một số ít nhất các tuyến đường mới để đảm bảo từ một điểm bất kỳ có thể đi tới điểm bất kỳ khác bằng tàu cao tốc.

Ví dụ, với  $n = 5$  và hiện có 4 đường:  $1 \rightarrow 2$ ,  $2 \rightarrow 3$ ,  $1 \rightarrow 4$  và  $4 \rightarrow 5$ . Để đảm bảo yêu cầu đã nêu, người ta cần xây dựng ít nhất 2 đường mới, chẳng hạn  $5 \rightarrow 3$  và  $3 \rightarrow 1$ .

**Yêu cầu:** Cho  $n$ ,  $m$  và các cặp  $(a, b)$  mô tả mạng giao thông sau khi đã sửa thành đường 1 chiều. Mỗi cặp  $(a, b)$  xác định tồn tại đường tàu  $a \rightarrow b$ . Hãy xác định số lượng tối thiểu các đường cần xây dựng thêm.

**Đữ liệu:** Vào từ file văn bản MONORAIL.INP:

- Dòng đầu tiên chứa 2 số nguyên  $n$  và  $m$ ,
- Mỗi dòng trong  $m$  dòng tiếp theo chứa 2 số nguyên  $a$  và  $b$ .

**Kết quả:** Đưa ra file văn bản MONORAIL.OUT một số nguyên – số đường mới.

**Ví dụ:**

MONORAIL.INP	MONORAIL.OUT
5 4 1 2 2 3 1 4 4 5	2

1 2
2 3
1 4
4 5

### \* Hướng dẫn thuật toán:

- Sử dụng Tajan để tìm các thành phần liên thông mạnh
- Mỗi thành phần liên thông mạnh thuộc 1 trong 3 loại sau:
  - + Loại 1: Thành phần liên thông chỉ có cung đi ra mà không có cung đi vào (ví dụ đỉnh 1 trong hình trên)
  - + Loại 2: Thành phần liên thông có cả cung đi ra và cả cung đi vào (ví dụ đỉnh 2 và 4 trong hình trên)
  - + Loại 3: Thành phần liên thông chỉ có cung đi vào mà không có cung đi ra (ví dụ đỉnh 3, 5 trong hình trên)
- Để tạo thành 1 vùng liên thông mạnh thì cần phải bổ sung cung nối từ thành phần liên thông loại 3 sang thành phần liên thông loại 1.

Tùy đó hình thành cách giải như sau:

- Đếm số thành phần liên thông loại 1 (gọi là d1) và loại 3 (gọi là d3)
- Kết quả của bài toán chính là  $\max(d1, d3)$ .