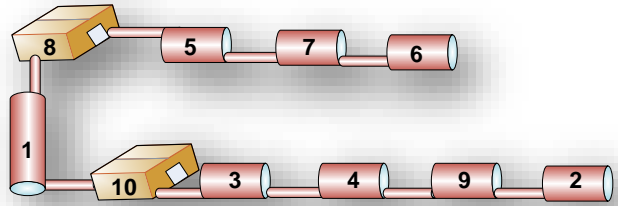


## BÀI TẬP DFS/BFS

### Sơ tán (EVA.\*)

Một Trung tâm nghiên cứu tuyệt mật (mà chúng ta không có quyền nói rõ tên ở đây) có  $n$  phòng thí nghiệm đặt ngầm trong lòng đất. Các phòng thí nghiệm được đánh số từ 1 đến  $n$  ( $1 \leq n \leq 10^5$ ). Giữa một số phòng có đường hầm nối với nhau, sao cho từ một phòng bất kỳ có thể đi đến phòng bất kỳ khác (có thể phải đi qua một số phòng nào đó). Độ dài mỗi đường hầm là như nhau và thời gian đi hết một đường hầm là 1. Không có đường hầm nào nối một phòng với chính nó, nhưng có thể có nhiều đường hầm cùng nối 2 phòng với nhau và tổng cộng trong Trung tâm có tất cả  $m$  đường hầm ( $1 \leq m \leq 10^5$ ). Đường hầm cho phép đi lại theo cả hai chiều. Có  $k$  phòng có lối thoát hiểm lên mặt đất ( $1 \leq k \leq n$ ). Trong trường hợp sơ tán khẩn cấp, tất cả các nhân viên phải tập trung ở những phòng có lối thoát hiểm.



**Yêu cầu:** Hãy xác định thời gian tối thiểu để nhân viên mỗi phòng tập trung về phòng có lối thoát hiểm trong trường hợp phải sơ tán khẩn cấp.

**Input:**

- Dòng đầu tiên chứa 2 số nguyên  $n$  và  $k$ ,
- Dòng thứ 2 chứa  $k$  số nguyên khác nhau cho biết các phòng có cửa thoát hiểm,
- Dòng thứ 3 chứa số nguyên  $m$ ,
- Mỗi dòng trong  $m$  dòng tiếp theo chứa 2 số nguyên xác định cặp phòng có đường hầm nối trực tiếp.

**Output:** Một dòng chứa  $n$  số nguyên, số thứ  $i$  xác định thời gian tối thiểu để nhân viên phòng  $i$  đi được tới phòng có lối thoát hiểm.

**Example:**

Input	Output
10 2 10 8 9 6 7 7 5 5 8 8 1 1 10 10 3 3 4 4 9 9 2	1 4 1 2 1 3 2 0 3 0

## Di chuyển robot (RMOVE.\*)

Cho một đồ thị có hướng gồm  $n$  đỉnh và  $m$  cung, hai con robot đứng ở đỉnh 1 và  $n$ . Hãy tìm cách di chuyển nhanh nhất hai con robot đến gặp nhau tại một đỉnh của đồ thị, biết rằng cả hai con robot chỉ được chạy theo các cung định hướng và không được dừng lại cho tới lúc gặp nhau tại một đỉnh nào đó. Thời gian robot đi qua một cung bất kỳ luôn là 1 đơn vị thời gian

*Input:*

- Dòng 1: Chứa hai số nguyên dương  $n \leq 1000; m \leq 5000$
- $m$  dòng tiếp theo mỗi dòng chứa hai số nguyên  $u, v$  tương ứng với cung  $(u, v)$  của đồ thị.

*Output:*

- Ghi thời gian tính từ lúc bắt đầu di chuyển đến khi hai robot gặp nhau (nếu không có phương án thì ghi -1)
- Trong trường hợp có phương án thực hiện thì dòng thứ hai ghi hành trình của robot thứ nhất theo thứ tự từ đỉnh 1 đến đỉnh gặp nhau và dòng thứ ba ghi hành trình của robot thứ hai với qui cách tương tự.

*Example:*

Input	Output
2 2 1 2 2 1	-1
4 4 1 3 3 2 4 1 1 2	2 1 3 2 4 1 2

## Tìm khớp và cầu (DFSGRAPH.\*)

Xét đơn đồ thị vô hướng  $G = (V, E)$  có  $n$  ( $1 \leq n \leq 3.10^5$ ) đỉnh, các đỉnh đánh số từ 1 đến  $n$  và  $m$  ( $1 \leq m \leq 5.10^5$ ) cạnh. Người ta định nghĩa một đỉnh là khớp nếu như bỏ đỉnh đó sẽ làm tăng số thành phần liên thông của đồ thị. Tương tự, một cạnh được gọi là cạnh cầu nếu xóa cạnh đó sẽ làm tăng số thành phần liên thông của đồ thị.

Vấn đề đặt ra là đếm tất cả các đỉnh khớp và cạnh cầu của đồ thị.

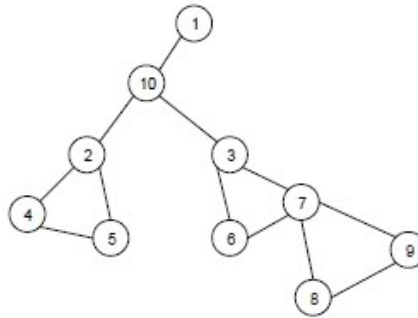
*Input:*

- Dòng đầu chứa hai số tự nhiên  $n, m$
- $m$  dòng tiếp theo, mỗi dòng chứa hai số nguyên  $u, v$  thể hiện có một cạnh nối đỉnh  $u$  với đỉnh  $v$  ( $1 \leq u, v \leq n; u \neq v$ )

*Output:* Gồm một dòng duy nhất chứa hai số nguyên, số thứ nhất - số lượng đỉnh khớp và số thứ hai - số lượng cạnh cầu của đồ thị

*Example:*

Input	Output
10 12	4 3
1 10	
10 2	
10 3	
2 4	
4 5	
5 2	
3 6	
6 7	
7 3	
7 8	
8 9	
9 7	



## Thành phố trung tâm (CAPITAL.\*)

Ở đất nước ByteLand có  $n$  thành phố (đánh số thứ tự từ 1 đến  $n$ ) được nối với nhau bằng  $m$  con đường hai chiều. Do yêu cầu về sự thịnh vượng của đất nước, cần thiết phải di chuyển thủ đô hiện thời về một thành phố nào đó trong số các thành phố nói trên. Quốc hội ByteLand muốn chọn một thành phố trung tâm nhất để di chuyển thủ đô đến đó.

Thành phố được gọi là trung tâm nếu như khi cấm các hoạt động giao thông qua thành phố này và trên các con đường nối với nó thì các thành phố còn lại tạo thành nhiều cụm thành phố nhất. Một cụm thành phố là một nhóm các thành phố mà giữa hai thành phố bất kỳ có thể đi đến được với nhau thông qua các thành phố trong cụm. Tất nhiên, nếu thêm một thành phố khác vào cụm thì tính chất trên bị phá vỡ.

Viết chương trình tìm thành phố trung tâm cho Quốc Hội

*Input:*

- Dòng đầu tiên ghi hai số nguyên dương  $n, m$  ( $2 \leq n, m \leq 300000$ )
- $m$  dòng tiếp theo, mỗi dòng ghi hai số  $a_i, b_i$  thể hiện có một đường đi hai chiều nối thành phố  $a_i$  với thành phố  $b_i$ .

*Output:* Ghi hai số  $C_{\max}$  và  $C$  thể hiện số cụm lớn nhất tạo ra khi cấm đường giao thông qua thành phố trung tâm và số hiệu của thành phố trung tâm này. Nếu có nhiều thành phố trung tâm thì chỉ in ra thành phố có chỉ số nhỏ nhất

*Example:*

Input	Output
5 5	3 2
1 2	
3 2	
4 2	
5 2	
1 3	

## Mạng máy tính (MNET.\*)

Một hệ thống  $N$  máy tính đánh số từ 1 đến  $N$  được nối mạng bằng  $M$  kênh truyền tin một chiều giữa một số cặp máy. Mạng cho gọi là thông suốt nếu như từ một máy  $u$  bất kỳ luôn có thể truyền

tin đến mỗi máy  $v$  trong số các máy còn lại hoặc theo kênh truyền tin từ  $u$  đến  $v$  hoặc thông qua một số kênh trung gian. Ta gọi một mạng con của mạng đã cho là một mạng gồm một số máy và các kênh nối chúng của mạng đã cho. Trong trường hợp mạng là không thông suốt nó sẽ phân rã thành một số mạng con thông suốt. Mạng con thông suốt được gọi là cực đại nếu như không tồn tại một mạng con thông suốt của mạng đã cho chứa nó như một mạng con. Bạn cần xác định số mạng con thông suốt cực đại của mạng đã cho.

*Input:*

- Dòng đầu tiên chứa hai số nguyên dương  $N, M$  ( $1 \leq N, M \leq 5 \cdot 10^5$ )
- Dòng thứ  $i$  trong  $M$  dòng tiếp theo ghi 2 số nguyên dương  $d_i, c_i$  cho biết kênh truyền tin thứ  $i$  cho phép truyền tin từ máy  $d_i$  sang máy  $c_i$ .

*Output:*

- Dòng đầu tiên ghi  $K$  là số mạng con thông suốt cực đại của mạng đã cho
- $K$  dòng tiếp theo, mỗi dòng ghi dãy các đỉnh thuộc cùng một mạng con thông suốt cực đại

Input	Output
9 14	3
1 2	1 2 3
1 4	4 5
1 7	6 7 8 9
2 3	
2 6	
3 1	
4 5	
5 4	
5 7	
6 7	
7 9	
9 6	
9 8	
8 9	

## Tham quan những cây cầu (BRIDTOUR.\*)

Quốc đảo ByteLand nổi tiếng trên thế giới về những cây cầu vượt biển của mình. Việc thực hiện các tour du lịch tham quan những cây cầu là một trong những niềm vui chính của du khách mỗi khi đến quốc đảo. Tuy nhiên theo Euler, người đã giải bài toán nổi tiếng "Bảy cây cầu ở Königsberg", thì không thể tham quan tất cả những cây cầu của quốc đảo này chỉ trong một tour du lịch được. Thay vào đó, mỗi du khách phải thực hiện nhiều tour du lịch mới hy vọng tham quan được hết những cây cầu này.

ByteLand có  $n$  hòn đảo xinh đẹp và có  $n-1$  cây cầu vượt biển nối các hòn đảo này với nhau sao cho từ một đảo bất kỳ luôn có thể đi đến hòn đảo bất kỳ khác chỉ bằng cách sử dụng các cây cầu này.

Lần đầu tiên đến ByteLand, Bờm rất muốn tham quan tất cả các cây cầu. Tuy nhiên, với bản tính hấp tấp vốn có của mình, anh ta chọn ngay  $m$  tour du lịch đầu tiên mà công ty du lịch quảng cáo

(vì Bờm chỉ có đủ tiền để đi  $m$  tour du lịch). Mỗi tour du lịch thực hiện việc tham quan tất cả các cây cầu từ đảo  $a$  đến đảo  $b$  nào đó.

Nộp tiền xong rồi, Bờm mới giật mình tự hỏi: "Có còn cây cầu nào mà mình không tham quan khi thực hiện xong  $m$  tour du lịch này nhỉ?".

Viết chương trình xác định số lượng các cây cầu mà Bờm không tham quan được.

*Input:*

- Dòng thứ nhất ghi số nguyên  $n$  - số hòn đảo ( $1 \leq n \leq 200000$ )
- $n - 1$  dòng tiếp theo: mỗi dòng gồm hai số nguyên  $a$  và  $b$  - có một cây cầu vượt biển nối đảo  $a$  và đảo  $b$ .
- Dòng thứ  $n + 1$  ghi số nguyên  $m$  - số tour du lịch mà Bờm đăng ký ( $1 \leq m \leq 200000$ )
- $m$  dòng tiếp theo: mỗi dòng gồm hai số nguyên  $a$  và  $b$  thể hiện một tour du lịch tham quan các cây cầu đi từ đảo  $a$  đến đảo  $b$ .

*Output:* Một số nguyên duy nhất thể hiện số cây cầu chưa được kiểm tra.

*Example:*

Input	Output
6	1
1 2	<i>Giải thích:</i> Tour thứ nhất Bờm tham quan các cây cầu (2, 3), (2, 4) và (4, 6). Tour thứ hai, Bờm tham quan các cây cầu (5, 4) và (4, 6). Cây cầu duy nhất chưa được tham quan là (1, 2).
2 3	
2 4	
4 5	
4 6	
2	
3 6	
5 6	

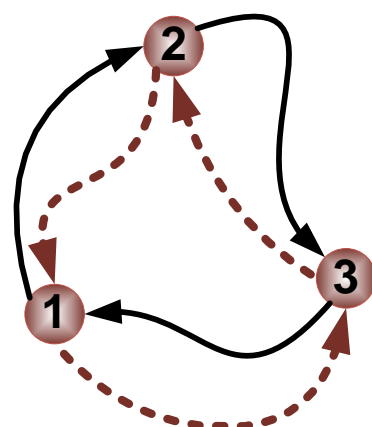
## Đường mới (NEWROADS.\*)

August IV là một quốc vương thông thái và công bằng. Ông chủ trương phát triển quan hệ thương mại giữa các thành phố trong nước.

Trong vương quốc có  $n$  thành phố và hiện có  $m$  đường thương mại. Mỗi đường thương mại nối trực tiếp 2 thành phố và là đường một chiều. Không có đường thương mại nào nối một thành phố với chính nó và không có hai đường thương mại nào có cùng thành phố xuất phát và kết thúc.

Thương nhân mang hàng hóa đi bán ở các thành phố khác và sẽ là tốt nhất nếu anh ta mua được hàng hóa mới và trở về thành phố của mình không phải với chiếc xe ngựa rỗng. Dĩ nhiên, khi đó vì mục đích an toàn, anh ta phải quay về bằng đường thương mại.

Nếu từ  $A$  có thể tới  $B$  theo các con đường thương mại và từ  $B$  có thể trở về  $A$  cũng theo các con đường thương mại thì  $A$  và  $B$  là hai thành phố thân thiện.



Để giảm bớt nỗi cực nhọc của các thương nhân bình dân, August cho sửa sang lại các tuyến thương mại hiện có và cho xây dựng thêm một số đường thương mại mới. Tuy vậy, với ngân khố quốc gia hạn chế, nhà vua quyết định chỉ mở thêm các đường thương mại nối các thành phố thân thiện và không trùng lặp với các đường hiện có, sao cho giữa các thành phố thân thiện luôn có thể đi được đến nhau bằng các đường nối trực tiếp.

**Yêu cầu:** Cho  $n$ ,  $m$  và mạng đường thương mại hiện có. Không có đường nào nối một thành phố với chính nó. Hãy xác định số đường mới tối thiểu cần xây dựng thêm.

**Input:**

- Dòng đầu tiên chứa 2 số nguyên  $n$  và  $m$  ( $1 \leq n \leq 100\,000$ ,  $0 \leq m \leq 200\,000$ ),
- Mỗi dòng trong  $m$  dòng tiếp theo chứa 2 số nguyên  $A$  và  $B$  xác định đường thương mại đi trực tiếp từ  $A$  đến  $B$ .

**Output:** Một số nguyên – kết quả tìm được.

**Example:**

Input	Output
3 3 1 2 2 3 3 1	3