



## Help the Prayatna pr team

Link: <http://www.spoj.com/problems/CAM5/>

Solution:

C++	<a href="http://ideone.com/xiNVED">http://ideone.com/xiNVED</a>
Java	<a href="https://ideone.com/yHxzJi">https://ideone.com/yHxzJi</a>
Python	<a href="https://ideone.com/DpTwsl">https://ideone.com/DpTwsl</a>

Python: (Mã nguồn tham khảo khi submit sẽ bị TLE do dữ liệu trên SPOJ chưa sẵn sàng phục vụ cho code python, các bạn có thể submit mã nguồn tại [www.algote.com](http://www.algote.com)).

**Tóm tắt đề:** Trong một nhóm người, có những người có thể quen biết nhau hoặc không. Những người quen biết nhau chia thành nhiều nhóm nhỏ. Xác định số lượng nhóm có trong nhóm người đó. Khi bạn biết được người nào quen biết người nào.

### Input

Dòng đầu tiên chứa số lượng bộ test  $t \leq 10$ . Mỗi bộ test bao gồm các thông tin:

Dòng đầu chứa số lượng người ( $2 \leq N \leq 100000$ ).

Dòng tiếp theo chứa số cặp người quen biết nhau ( $0 \leq e \leq N/2$ ).

e dòng tiếp theo chứa các cặp mỗi cặp là 2 số a, b đại diện cho người a quen với người b.

### Output

Mỗi bộ test. In số nhóm có trong N người ban đầu.

### Ví dụ

2  4 2 0 1 1 2  3 0	2 3
---	--------

**Giải thích:** Có 2 bộ test.

Bộ đầu tiên có 4 người. Người 0 quen người 1, người 1 quen người 2. Còn người 3 không quen ai. Vậy ta có 2 nhóm nhóm 0, 1, 2 và nhóm 3. Kết quả xuất ra giá trị 2.

Bộ test thứ 2 có 3 người và không ai là bạn của ai cả. In có 3 nhóm và in ra giá trị 3.

**Hướng dẫn giải:** Lần lượt chạy DFS cho các đỉnh, nếu mỗi lần chạy vẫn còn đỉnh nào có giá trị false nghĩa là vẫn còn nhóm nào đó khác nhóm trước đã chạy. Bạn sẽ tăng biến đếm lên. Khi chạy hết N đỉnh thì kết quả cuối cùng là biến đếm.

**Độ phức tạp:**  $O(T * (V + E))$  với T là số lượng bộ dữ liệu,  $O(V + E)$  là độ phức tạp của thuật toán DFS với V là số lượng đỉnh của đồ thị và E là số lượng cạnh của đồ thị.

Big-O Coding