**Ví dụ**3 2

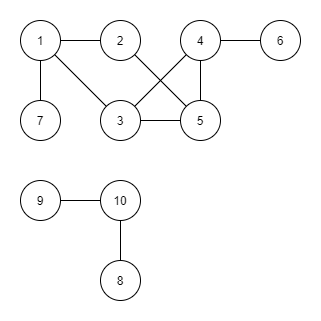
1 2

2 3

* 1 là bạn của 2, 2 là bạn của 3 ⇒ 1, 2, 3 là bạn của nhau

⇒ Chỉ có 1 nhóm bạn và nhóm có 3 người

⇒ Số lượng người của nhóm bạn đông nhất là 3



10 12

1 2

3 1

3 4

5 4

3 5

4 6

5 2

2 1

7 1

1 2

9 10

8 9

* Ở đây, giữa 2 đỉnh A và B có cạnh khi A và B là bạn của nhau.
* Có 2 nhóm bạn: 1 nhóm có 7 người và 1 nhóm có 3 người.
* Số lượng người của nhóm đông nhất là 7.

## **Nhận xét:**

* Cứ 2 người là bạn của nhau thì cho vào cùng 1 set.
* Để biết nhanh được 1 nhóm bạn có bao nhiêu người: Dùng thêm mảng cnt[i] là số lượng người trong nhóm i làm đại diện. Khi kết hợp 2 nhóm lại với nhau phải cập nhật lại cnt của đỉnh tương ứng.

## **Giải thuật:**

B1: Đọc dữ liệu. Khởi tạo parent[i] = i và cnt[i] = 1 cho tất cả đỉnh i.

B2: Với mỗi cặp đỉnh u, v là bạn của nhau, thêm 2 đỉnh vào cùng 1 nhóm bằng cách:

* Tìm đỉnh đại diện của u và v (up và vp)
* Nếu 2 đỉnh u và v chưa cùng nhóm (vp != up):
  + Cho parent[up] = vp
  + Cập nhật: cnt[vp] += cnt[up]

B3: Duyệt qua mảng cnt và xuất giá trị lớn nhất.

## **Mã giả**

findSet(u):

while u != parent[u]:

u = parent[u]

return u

unionSet(u, v):

up = findSet(u)

vp = findSet(v)

if up != vp:

parent[up] = vp

cnt[vp] += cnt[up]

main():

read(test)

for t = 0 to test - 1:

read(N, M)

parent = [] \* (N + 1)

for i = 1 to N:

parent[i] = i

cnt = [1] \* (N + 1)

for i = 0 to M - 1:

read(u, v);

unionSet(u, v) //O(N)

ans = 0

for i = 1 to N:

ans = max(ans, cnt[i])

print(ans)

Độ phức tạp: O(T \* N \* M)