<https://www.hackerearth.com/practice/algorithms/graphs/breadth-first-search/practice-problems/algorithm/containers-of-choclates-1/description/>

# Chiến tranh giữa các bộ tộc (War)

Tại vùng đất ByteLand, chiến tranh giữa các bộ tộc đang diễn ra. Hai bộ tộc Raikage và Zetsu đang chiến đấu với nhau. Cuộc chiến rất khốc liệt và chưa biết ai thắng ai. Bộ tộc Leaf là láng giềng của họ, và Leaf muốn biết bộ tộc trong số hai bộ tộc trên sẽ sống sót.

Bộ tộc Leaf đã lên một kế hoạch do thám và thu thập thông tin từ hai bộ tộc Raikage và Zetsu. Leaf đã biết thông tin về tất cả các trận đấu tay đôi. Trận đấu tay đôi là trận đấu giữa một thành viên của Raikage và một thành viên của Zetsu.

Mỗi thành viên của Raikage cũng như của Zetsu đều được gán một nhãn ID duy nhất. Bộ tộc Leaf biết ID của các cuộc đấu tay đôi, nhưng không biết ID nào thuộc về Raikage hay Zetsu.

Trên thực tế, họ chỉ muốn đoán biết số lượng binh sĩ tối đa có thể của Raikage. Nhưng vì cư dân của Leaf không giỏi lập trình nên họ nhờ bạn giúp đỡ. Hãy giúp Leaf tìm câu trả lời.

### Đầu vào

Dòng đầu tiên của đầu vào chứa một số nguyên *T*, là số bộ dữ liệu kiểm tra. Tiếp theo là *T* bộ dữ liệu, mỗi bộ dữ liệu gồm:

Mỗi trường hợp chứa một số nguyên N biểu thị số lượng các trận đánh kép. Mỗi dòng N tiếp theo sẽ chứa hai số nguyên A, B khác nhau biểu thị có một cuộc chiến giữa những người lính có id A và B.

### Đầu ra

Đối với mỗi trường hợp, in số trường hợp và các thành viên (binh sĩ) tối đa có thể của Quân đội Raikage.

### Ràng buộc

* 1 ≤ *T* ≤ 20
* 1 ≤ *N* ≤ 105
* 1 ≤ *A, B* ≤ 2\*104 (*A* và *B* là ID của hai đối thủ khác bộ tộc)

### Ví dụ

|  |  |
| --- | --- |
| Đầu vào | Đầu ra |
| 4  4  1 2  2 3  2 4  2 5  6  2 6  1 5  3 4  7 9  5 8  4 10  1  100 200  4  1 2  2 3  3 4  4 5 | Case 1: 4  Case 2: 6  Case 3: 1  Case 4: 3 |

### Giải thích ví dụ

* Bộ dữ liệu 1: {2} có thể thuộc về zetsu và {1,3,4,5} có thể thuộc về raikage ... Vì vậy, số lượng tối đa có thể của raikage là 4.
* Bộ dữ liệu 2: {2,5,4,7} có thể thuộc về zetsu và {6,1,3,9,8,10} có thể thuộc về raikage .... Vì vậy, số lượng tối đa có thể của raikage là 6.
* Bộ dữ liệu 4: {2,4} có thể thuộc về zetsu và {1,3,5} có thể thuộc về raikage ....

### Solution

Chúng ta có một tập *N* cặp nút. Ta cần tìm số nút tối đa (gọi nó là tập M) sao cho có chính xác 1 phần tử từ mỗi cặp. Ví dụ: nếu các cặp đã cho là: (1, 2), (2, 3), (3, 4) thì lựa chọn tối đa sẽ là {1,3} hoặc {2, 4}

Giải pháp: Sử dụng BFS để tính M.

**Hoạt động 1**: Ban đầu ta tạo ra một mảng các nút với mỗi nút là một SET. Tương ứng với mảng *set*, ta tạo một mảng *visit* để kiểm tra xem một nút đã được thăm hay hay chưa. Ban đầu *visit* được gán là *false*. Sau đó, ta tạo ra một mảng *isthere* để theo dõi các nút xuất hiện ít nhất một lần trong cặp bất kỳ nào. *Isthere [i] = 1* → *i* có mặt trong tập hợp

**Hoạt động 2**: Với mỗi cặp (*a, b*), ta thêm cạnh nối giữa *a* và *b,* tức là đẩy *a* vào *set[b]* và đẩy *b* vào *set[a]*. Đồng thời, ta gán *visit[a]* = *visit[b] = false* để đánh dấu chúng là chưa được thăm để phục vụ cho hoạt động 3. Đồng thời ta cũng gán *isthere[a] = isthere [b] = 1* biểu thị *a, b* đã xuất hiện trong danh sách các cặp.

**Hoạt động 3**: Bây giờ ta áp dụng BFS để duyệt và đếm các nút. Giả sử có một cạnh *a-b*, ta có thể chọn *a* hoặc *b*.

Để một thành phần liên thông có 4 cấp (0, 1, 2, 3) ta có thể chọn các phần tử cấp 0 và 2 (Gọi nó là LV0) hoặc các phần tử của cấp 1 và 3 (Gọi nó là LV1). Cuối cùng, ta lấy max(LV0, LV1) của mỗi thành phần liên thông cụ thể. Tổng của tất cả các *max* từ tất cả các thành phần liên thông cho ta câu trả lời bài toán.

Chúng ta đẩy một nút vào hàng đợi bằng thẻ của nó (giả sử là LV0) trong một cặp. Tương ứng với nút mà ta đã đẩy, ta đẩy tất cả các nút kề với nó bằng thẻ khác (tức là LV1) và cứ thế ta đi qua đồ thị.

Ta có thể biểu diễn thẻ này bằng một biến nguyên *ch* và tăng nó ở mọi cấp độ. Vì *ch % 2* = (ch + 2) % 2...

Lưu ý: Sử dụng kỹ thuật Fast IO để nhập và xuất nhanh hơn.