OJ 1197

The Suspects

Prof. Edson Alves – UnB/FCTE

Problema

Severe acute respiratory syndrome (SARS), an atypical pneumonia of unknown aetiology, was recognized as a global threat in mid-March 2003. To minimize transmission to others, the best strategy is to separate the suspects from others.

In the Not-Spreading-Your-Sickness University (NSYSU), there are many student groups. Students in the same group intercommunicate with each other frequently, and a student may join several groups.

To prevent the possible transmissions of SARS, the NSYSU collects the member lists of all student groups, and makes the following rule in their standard operation procedure (SOP).

Once a member in a group is a suspect, all members in the group are suspects.

However, they find that it is not easy to identify all the suspects when a student is recognized as a suspect. Your job is to write a program which finds all the suspects.

Entrada e saída

Input

The input file contains several cases. Each test case begins with two integers n and m in a line, where n is the number of students, and m is the number of groups. You may assume that $0 < n \leq 30000$ and $0 \leq m \leq 500$. Every student is numbered by a unique integer between 0 and $n - 1$, and initially student 0 is recognized as a suspect in all the cases. This line is followed by m member lists of the groups, one line per group. Each line begins with an integer k by itself representing the number of members in the group. Following the number of members, there are k integers representing the students in this group. All the integers in a line are separated by at least one space.

A case with $n = 0$ and $m = 0$ indicates the end of the input, and need not be processed.

Output

For each case, output the number of suspects in one line.

Exemplo de entradas e saídas

Sample Input

```
100 4
2 1 2
5 10 13 11 12 14
2 0 1
2 99 2
200 2
1 5
5 1 2 3 4 5
1 0
0 0
```

Sample Output

```
4
1
1
```

Solução com complexidade $O(NM \log NM)$

- A relação “teve contato com um suspeito” é uma relação de equivalência
  (a) O suspeito teve contato consigo mesmo (reflexiva)
  (b) Se o suspeito teve contato com uma pessoa, a pessoa teve contato com o suspeito (simétrica)
  (c) Se o suspeito teve contato com uma pessoa, e a pessoa teve contato com fulano, é considerado que fulano também teve contato com o suspeito (transitiva)
- Assim, para identificar o grupo de pessoas que teve contato com o suspeito, basta usar uma UFDS
- A resposta será o número de elementos do conjunto que contém o suspeito
- A complexidade da solução, no pior caso, é $O(NM \log NM)$

Solução com complexidade $O(NM \log NM)$

```cpp
#include <bits/stdc++.h>

using namespace std;

class UFDS
{
    std::vector<int> size, ps;

public:
    UFDS(int N) : size(N + 1, 1), ps(N + 1)
    {
        std::iota(ps.begin(), ps.end(), 0);
    }

    int find_set(int x)
    {
        return x == ps[x] ? x : (ps[x] = find_set(ps[x]));
    }

    bool same_set(int x, int y) { return find_set(x) == find_set(y); }
```

Solução com complexidade $O(NM \log NM)$

```
    void union_set(int x, int y)
    {
        if (same_set(x, y))
            return;

        int p = find_set(x);
        int q = find_set(y);

        if (size[p] < size[q])
            std::swap(p, q);

        ps[q] = p;
        size[p] += size[q];
    }

    int get_size(int x) { return size[find_set(x)]; }
};
```

Solução com complexidade $O(NM \log NM)$

```
int main()
{
    ios::sync_with_stdio(false);

    int n, m;

    while (cin >> n >> m, n | m)
    {
        UFDS ufds(n);

        while (m--)
        {
            int k;
            cin >> k;

            if (k == 0) continue;

            int x;
            cin >> x;
            --k;
```

Solução com complexidade $O(NM \log NM)$

```
            while (k--)
            {
                int y;
                cin >> y;

                ufds.union_set(x, y);
            }
        }

        cout << ufds.get_size(0) << '\n';
    }

    return 0;
}
```