

<https://www.acmicpc.net/problem/1043>



지민이는 사건의 진실을 알고 있는 사람에게는 과장된 얘기를 안 할 것입니다.  
이때 지민이가 과장된 얘기를 할 수 있는 파티 수의 최댓값을 구하는 문제입니다.

## 풀이

진실을 아는 사람들이 속한 파티는 일단 진실을 말해야 하니 그 파티에 있는 사람들은 진실을 알고 있습니다.

처음에 입력받은 진실을 아는 사람들을 기준으로 먼저 진실을 말하고 그 파티에 속한 사람들 기준으로 다시 그 사람들이 속한 파티에 진실을 말해야 합니다.

이 과정을 반복하다 보면 진실을 말해야 하는 파티의 수를 구할 수 있습니다.

다른 사람들의 실수를 보니 파티의 순서를 고려하시는 것 같은데 순서는 관계없습니다.

진실을 알아야 하는 사람들끼리 연결하다 보면 답을 구할 수 있습니다.

구현은 진실을 아는 사람들끼리 연결한다고 생각하면 플로이드 워셜을 사용해도 될 것 같고 그냥 시뮬레이션으로 구현해도 될 것 같습니다.

방법이 여러 가지가 있는 것 같은데 저는 그래프 탐색(BFS)로 풀었습니다.

편하신 방법으로 구현하세요!

소스 코드

```

#include <bits/stdc++.h>
#define FIO ios::sync_with_stdio(0), cin.tie(0), cout.tie(0)
#define FOR(i,a,b) for(int i=a; i<=b; i++)
using namespace std;

int main(){
    FIO;
    int n,m; cin >> n >> m;
    queue<int> q;
    bool t[55] = {}, a[55][55] = {};
    int k; cin >> k;
    while(k--){
        int x; cin >> x;
        t[x] = 1;
        q.emplace(x);
    }
    FOR(i,1,m){
        int x; cin >> x;
        while(x--){
            int y; cin >> y;
            a[i][y] = 1;
        }
    }
    while(q.size()){
        int now = q.front(); q.pop();
        FOR(i,1,m) if(a[i][now]){
            FOR(j,1,n) if(!t[j] && a[i][j]){
                t[j] = 1;
                q.emplace(j);
            }
        }
    }
    int ans = 0;
    FOR(i,1,m){
        bool f = 0;
        FOR(j,1,n){
            if(a[i][j] && t[j]){
                f = 1;
                break;
            }
        }
        if(!f) ans++;
    }
    cout << ans << '\n';
    return 0;
}

```