// algoleague
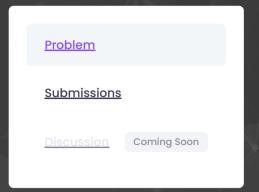
Run Roka Run

⌂ • Contest List • Algorithm Competition Summer Camp 2023 Foundation Upsolving Contest • Problem List • Run Roka Run • Problem

Problem

Submissions

Discussion   Coming Soon

All members of inzva gathered in Beykoz for summer camp. Roka, Utku's dog, got very excited when she saw them. And she started to look for Utku to tell him that. But inzva Sanctuary is very big and complex. There are a lot of corridors between the rooms. You are given 4 integers; n, number of rooms and m, number of corridors in the sanctuary with r and u, the room numbers of Roka and Utku. Rooms are numbered from 1 to n. You are also given adjacency lists of all rooms, each indicates a list of rooms that are adjacent to a room. Can you tell us at least how many corridors Roka needs to go through to reach Utku?

Notes: Each pair of rooms can share at most one corridor and there are no intersecting corridors (inzva Sanctuary is a magical place). That's why you need to go through a room to change the corridor. And each room is reachable from every other room.

Input Format

The first line contains two space-separated integers n and m — the number of rooms and corridors in the sanctuary, respectively.

The second line contains two space-separated integers r and u — the room numbers of Roka and Utku, respectively.

Each of the next n lines contains at least three space-separated integers. First integer a, indicates a room and it is different in each line. Second integer b, indicates the number of room(s) that is/are adjacent to a, followed by room numbers.

Output Format

Print one integer, minimum number of corridors between Roka and Utku.

Constraints

- $2 \leq n \leq 1000$
- $n - 1 \leq m \leq \frac{n \cdot (n-1)}{2}$

Sample Input 1 ⎘

```
10 12
5 9
1 3 5 6 8
2 3 3 7 10
3 3 2 5 6
4 2 5 7
5 2 1 3
6 3 1 3 9
7 2 2 4
8 1 1
9 2 6 10
10 2 2 9
```

Sample Output 1 ⎘

```
3
```

```cpp
#include <bits/stdc++.h>
using namespace std;
vector <int> adj[1001];
bool visited[1001];


int bfs(int start, int end){
    memset(visited, 0, sizeof(visited));
    queue<pair<int,int> > q;
    q.push({start, 0});
    visited[start] = true;

    while(!q.empty()){
        int v = q.front().first;
        int dist = q.front().second;
        q.pop();

        if(v == end)
            return dist;

        for(int u : adj[v]) {
            if(!visited[u]){
                q.push({u, dist + 1});
                visited[u] = true;
            }
        }
    }
    return -1;
}
int main() {
    int n, m;
```

Upload File   Test against custom test case   Run Code   Submit

✔ Sample Test Case 0

Accepted

Input(stdin)

```
1   10 12
2   5 9
3   1 3 5 6 8
4   2 3 3 7 10
5   3 3 2 5 6
6   4 2 5 7
7   5 2 1 3
8   6 3 1 3 9
9   7 2 2 4
```

Output(stdin)

```
1   3
2
```

Expected Output

```
1   3
```