All Contests > Assignment 02 | Introduction to Algorithms | Batch 03 > Know Your Level

Know Your Level

Problem Submissions Leaderboard Discussions

Problem Statement

Zaman lives in a city of pizzas where every street has several pizza shops and everyone loves pizza there. Suppose there are **N** pizza shops in Zaman's area. All pizza shops have unique numbers written in the shop, the number of the first pizza shop in his city is from **0** to **10^5**. There are **E** roads between pizza shops, and these pizza shops form a undirected connected graph where you can divide them in levels where the level starts from **0**. Zaman lives in level **L**, now he wants to know which pizza shops are there in his level. Can you help him to get the desired pizza shops?

Note: If there are no pizza shops at level **L**, then print **-1**.

Input Format

- First line will contain two integers **N** and **E**, number of pizza shops and roads repectively.
- The next **E** lines will contain two integeres **A** and **B**, which means there is a road between A and B.
- The last line will contain **L**, the level where Zaman lives.

Constraints

1. 0 < **N** <= 10^5
2. 1 <= **E** <= 10^5
3. 0 <= **A, B, L** <= 10^5

Output Format

- Output the numbers written on the pizza shops that are in level **L** in **ascending order**.

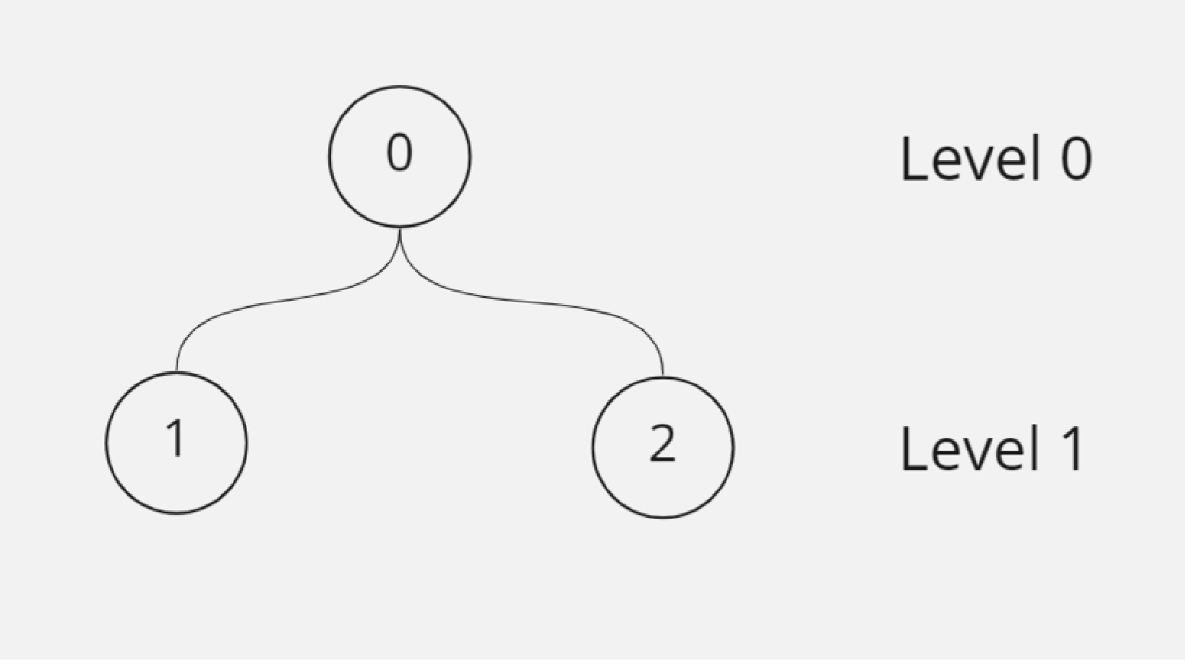
Sample Input 0

```
3 2
0 1
0 2
1
```

Sample Output 0

```
1 2
```

Explanation 0



Submissions: 201
Max Score: 20
Difficulty: Easy

Rate This Challenge:
☆☆☆☆☆

More

C++20

```
#include<bits/stdc++.h>
using namespace std;

const int N = 1e5;
vector<int> adj[N];
bool visited[N];
int level [N];


void bfs(int src)
{
    queue<int> q;
    q.push(src);
    visited[src] = true;
    level[src] = 0;

    while(!q.empty())
    {
        int u = q.front();
        q.pop();
        for(int v : adj[u])
        {
            if(visited[v] == false)
            {
                q.push(v);
                visited[v] = true;
                level[v] = level[u]+1;
            }
        }
    }

}

int main()
{

    int n, m;
    cin >> n >> m;
    for(int i = 0; i<m; i++)
    {
        int u, v;
        cin >> u >> v;
        adj[u].push_back(v);
        adj[v].push_back(u);

    }
    int  L; cin >> L;


    bfs(0);
    vector<int> vec(n);
    for(int i =0;i<n; i++)
    {
        if(level[i]==L)
        {
            // cout<<i<<" ";
            vec.push_back(i);
        }
        // cout<<"Level of "<<i<<": "<<level[i]<<endl;
    }

    // sort(vec.begin(),vec.end());
    for(auto element : vec)
    {
        if(element)
        cout<<element<<" ";
        else cout<<-1<<endl;
    }
    cout<<endl;




    return 0;
}

```

Line: 1 Col: 1

Upload Code as File □ Test against custom input

Run Code Submit Code

Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy |