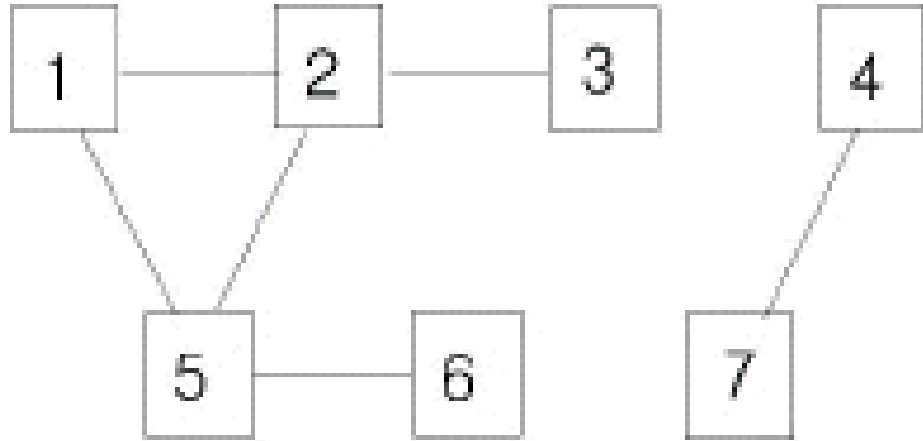


-2606 바이러스-

202021006 소프트웨어학과 김예진

2606 – 바이러스



< 그림 1 >

그림과 같이 컴퓨터가 연결되어 있을 때 1번의 컴퓨터가 바이러스에 걸렸을 때, 총 몇 대의 컴퓨터가 바이러스에 걸리는지 알아내는 문제

Ex)

1번 -> 바이러스

- 1번과 연결되어 있는 2번, 5번 컴퓨터 바이러스

2번, 5번 -> 바이러스

- 2번과 연결되어 있는 4번 컴퓨터 바이러스

- 5번과 연결되어 있는 6번 컴퓨터 바이러스

따라서 2, 3, 5, 6의 컴퓨터가 바이러스에 걸림

접근 방법

- 1) 일단 문제에 나와있는 그림을 보고 그래프라고 생각.
- 2) 연결되어 있는 컴퓨터 들을 모두 탐색해야 하므로 dfs를 이용하여 풀이.
- 3) 2차원배열을 선언하여 연결되어 있는 각각의 컴퓨터들을 1로 저장 (나머지는 0)
- 4) Dfs함수를 선언하고 이미 방문했던 컴퓨터번호인지 확인하기 위해 visit 라는 배열을 선언 (1번 컴퓨터는 무조건 1)
- 5) Dfs가 시작될때 결과값을 나타내는 res값을 +1해준다.
- 6) For 반복문을 이용하여, 그래프[1][i]가 1이고 visit 가 0이면 재귀호출.

코드 설명

```
#include <iostream>
using namespace std;

int graph[101][101] = {0};
int visit[101] = {0};
int res = 0;
```

-> graph : 그래프를 행렬로 표현하기 위한 2차원 배열

-> visit : dfs를 할 때 노드의 방문여부를 확인하기 위한 배열

-> res : 결과값을 나타내는 변수

```
int dfs(int start, int net){
    visit[start] = 1;
    res++;

    for (int i = 0; i <= net; i++){
        if(graph[start][i] == 1 && visit[i] == 0)
            dfs(i, net);
    }

    return res;
}
```

-> 1번 컴퓨터는 무조건 방문

-> 결과값 + 1

-> 각 행을 확인하면서 바이러스에 걸리는 컴퓨터의 개수를 센다.

코드 설명

```
int main(void){
    int num, net;

    cin >> num >> net;

    for (int i = 0; i < net; i++){
        int u, v;

        cin >> u >> v;
        graph[u][v] = graph[v][u] = 1;
    }

    cout << dfs(1, num) - 1 << '\n';

    return 0;
}
```

-> num : 컴퓨터의 개수

-> net : 네트워크 상에서 직접 연결되어 있는 컴퓨터 쌍의 수

-> 연결되어 있는 컴퓨터들을 행렬에 1로 저장해준다.

-> 결과값을 1도 포함되어 있기 때문에, 출력할때는 1을 빼고 출력한다.

