

# 2022 여름학기 동국대학교 SW역량강화캠프

## 9일차. 그래프 1

## ● 그래프 저장 방법 (인접 리스트)

- ▶ 그래프 탐색 문제에서 “x와 연결된 정점”을 빠르게 찾을 수 있는 그래프 저장 형태
- ▶ x번 List에 x와 연결된 정점들을 저장하는 형태

```
graph = new ArrayList<ArrayList<Integer>>();
for(int i=0; i<=N ; i++) {
    graph.add(new ArrayList<Integer>());
}

for(int i = 0; i < M; i++) {
    st = new StringTokenizer(br.readLine());
    int u = Integer.parseInt(st.nextToken());
    int v = Integer.parseInt(st.nextToken());

    graph.get(u).add(v);
    graph.get(v).add(u);
}
```

```
int N,M,V; cin>>N>>M>>V;
for(int i=0; i<M; i++)
{
    int x,y; cin>>x>>y;
    v[x].push_back(y);
    v[y].push_back(x);
}
```

### ● DFS와 BFS (4321)

#### | 문제

그래프를 DFS로 탐색한 결과와 BFS로 탐색한 결과를 출력하는 프로그램을 작성하시오. 단, 방문할 수 있는 정점이 여러 개인 경우에는 정점 번호가 작은 것을 먼저 방문하고, 더 이상 방문할 수 있는 점이 없는 경우 종료한다. 정점 번호는 1번부터 N번까지이다.

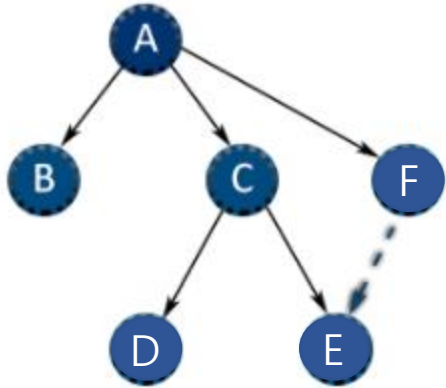
#### | 입력

첫째 줄에 정점의 개수  $N(1 \leq N \leq 1,000)$ , 간선의 개수  $M(1 \leq M \leq 10,000)$ , 탐색을 시작할 정점의 번호  $V$ 가 주어진다. 다음 M개의 줄에는 간선이 연결하는 두 정점의 번호가 주어진다. 어떤 두 정점 사이에 여러 개의 간선이 있을 수 있다. 입력으로 주어지는 간선은 양방향이다.

#### | 출력

첫째 줄에 DFS를 수행한 결과를, 그 다음 줄에는 BFS를 수행한 결과를 출력한다. V부터 방문된 점을 순서대로 출력하면 된다.

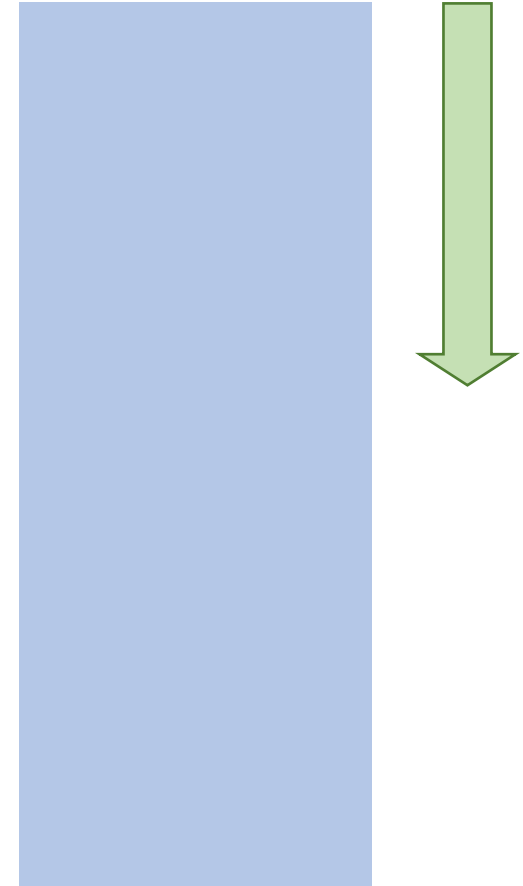
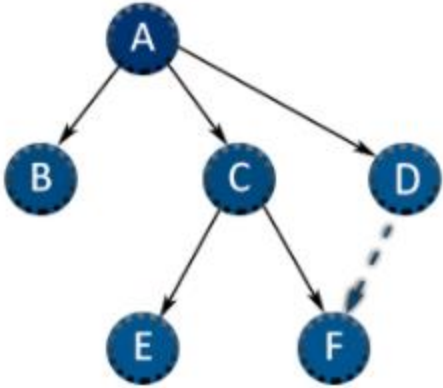
### ▶ DFS 정점의 탐색 순서를 출력



--	--	--	--	--	--

```
public static void dfs(int x) {  
    visit[x] = true;  
    sb.append(x+" ");  
    for(int y : graph.get(x)) {  
        if(!visit[y]) {  
            dfs(y);  
        }  
    }  
}
```

### ▶ BFS 정점의 탐색 순서를 출력



```
Queue<Integer> queue = new LinkedList<>();
visit = new boolean[N + 1];

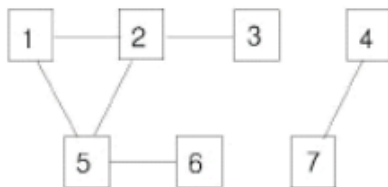
queue.offer(V); // 큐에 처음 점 V를 offer
visit[V] = true; // visit 배열은 방문 여부보다는 큐에 들어간 적 있는지 여부를 확인한다.(중복 offer 방지)
while (!queue.isEmpty()) {
    int x = queue.poll(); // queue에서 꺼낸 값 x
    sb.append(x + " "); // x를 출력
    for (int y : graph.get(x)) { // x와 인접한 y
        if (!visit[y]) { // y를 큐에 넣은 적이 없다면
            queue.offer(y); // y를 queue에 넣고 visit에 체크
            visit[y] = true;
        }
    }
}
System.out.println(sb.toString());
```

### ● 바이러스 (877)

#### 문제

신종 바이러스인 웜 바이러스는 네트워크를 통해 전파된다. 한 컴퓨터가 웜 바이러스에 걸리면 그 컴퓨터와 네트워크 상에서 연결되어 있는 모든 컴퓨터는 웜 바이러스에 걸리게 된다.

예를 들어 7대의 컴퓨터가 <그림 1>과 같이 네트워크 상에서 연결되어 있다고 하자. 1번 컴퓨터가 웜 바이러스에 걸리면 웜 바이러스는 2번과 5번 컴퓨터를 거쳐 3번과 6번 컴퓨터까지 전파되어 2, 3, 5, 6 네 대의 컴퓨터는 웜 바이러스에 걸리게 된다. 하지만 4번과 7번 컴퓨터는 1번 컴퓨터와 네트워크 상에서 연결되어 있지 않기 때문에 영향을 받지 않는다.



< 그림 1 >

어느 날 1번 컴퓨터가 웜 바이러스에 걸렸다. 컴퓨터의 수와 네트워크 상에서 서로 연결되어 있는 정보가 주어질 때, 1번 컴퓨터를 통해 웜 바이러스에 걸리게 되는 컴퓨터의 수를 출력하는 프로그램을 작성하시오.



- ▶ 재귀함수를 이용한 풀이
- ▶ dfs 함수를  $\text{dfs}(x)$  :  $x$ 에서 갈 수 있는 모든 정점의 `visited`를 `true`로 바꾸는 함수
- ▶ dfs를 통해서 모든 갈 수 있는 정점들을 탐색

```
public class main {  
    static ArrayList<ArrayList<Integer>> Graphlist;  
    static boolean visited[];  
  
    public static void main(String[] args) throws IOException {  
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  
        StringTokenizer st;  
        int N = Integer.parseInt(br.readLine());  
        int M = Integer.parseInt(br.readLine());  
  
        Graphlist = new ArrayList<ArrayList<Integer>>();  
        visited = new boolean[N + 1];  
  
        // Graph 인접리스트 만들기  
        for (int i = 0; i <= N; i++)  
            Graphlist.add(new ArrayList<Integer>());  
        for (int i = 0; i < M; i++) {  
            st = new StringTokenizer(br.readLine());  
            int p = Integer.parseInt(st.nextToken());  
            int q = Integer.parseInt(st.nextToken());  
            Graphlist.get(p).add(q);  
            Graphlist.get(q).add(p);  
        }  
    }  
}
```

```
dfs(1);

int ans = 0;
for (int i = 2; i <= N; i++) {
    if (visited[i]) {
        ans++;
    }
}
System.out.println(ans);
}

public static void dfs(int x) {
    visited[x] = true;
    for (int y : Graphlist.get(x)) {
        if (visited[y] == false) {
            dfs(y);
        }
    }
}
```

### ● 연구활동 가는 길(3003)

#### | 문제

정올이는 GSHS에서 연구활동 때문에 교수님을 뵈러 A대학교를 가려고 한다.

출발점과 도착점을 포함하여 중간 지역은  $n$ 개이고, 한 지역에서 다른 지역으로 가는 방법이 총  $m$  개이다.

각 방법으로 지역 사이를 이동하는데는 어느 정도 비용이 필요하고, 한 지역에서 다른 지역으로 가는 어떠한 방법이 존재하면 같은 방법과 비용을 통해 역방향으로 갈 수 있다.

GSHS는 지역 1이고 A대학교는 지역  $n$ 이라고 할 때 대학까지 최소 비용을 구하여라.

$n$ 은 10 이하,  $m$ 은 30 이하, 그리고 한 지역에서 다른 지역으로 가는 데에 필요한 비용은 200 이하 자연수이며 정점  $a \rightarrow$  정점  $b$ 로의 간선이 여러 개 있을 수 있으며, 자기 자신으로 가는 간선을 가질 수도 있다.

- ▶ DFS 완전탐색을 이용한 풀이
- ▶ dfs 함수를  $\text{dfs}(x, \text{dist})$  : 정점  $x$ 에 거리  $\text{dist}$ 로 도달한 상태, 인접한 정점들을 탐색하는 함수
- ▶ dfs를 통해서 모든 갈 수 있는 정점들을 탐색

```
void dfs(int x,int dist){ // x의 위치에 거리 dist로 도착

    if(x == N){
        mini=min(dist,mini);
        return;
    }

    for(auto i:v[x])
    {
        if(!chk[i.first])
        {
            chk[i.first] = true;
            dfs(i.first, dist+i.second);
            chk[i.first] = false;
        }
    }
}
```

```
cin>>N>>M;
for(int i=0;i<M;i++){
    cin>>x>>y>>z;
    v[x].push_back({y,z});
    v[y].push_back({x,z});
}
chk[1] = true;
dfs(1,0);
cout<<mini;
```