



1260 - DFS와 BFS

난이도 실버 2

문제

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.

풀이과정

DFS와 BFS 알고리즘의 단순 구현 문제였다. 그래프를 인접리스트로 정리한 뒤 오름차순으로 정렬하고 DFS와 BFS 각각을 구현한 함수를 돌려 방문 순서를 출력해주었다.

먼저 DFS 함수 코드는 다음과 같다.

```
static void dfs(int num) {  
  
    // 방문 체크  
    visited[num] = 1;  
  
    System.out.print(num + " ");  
  
    int idx = 0;  
    for(int node : graph[num]){  
        if(visited[graph[num].get(idx++)] < 1)  
            dfs(node);  
    }  
  
}
```

재귀적으로 구현했으며, 방문여부를 꼭 체크해주었다.

BFS 함수 코드는 다음과 같다.

```
static void bfs(int num) {  
  
    StringBuilder sb = new StringBuilder();  
    Queue<Integer> q = new LinkedList<>();  
  
    q.offer(num);  
  
    visited[num] = 1;  
  
    while(!q.isEmpty()){  
        int node = q.poll();  
        sb.append(node + " ");  
        for(int i=0 ; i<graph[node].size() ; i++){  
            int tmp = graph[node].get(i);  
            if(visited[tmp] < 1){  
                visited[tmp] = 1;  
                q.offer(tmp);  
            }  
        }  
    }  
  
    System.out.println(sb.toString());  
  
}
```

큐를 이용하여 구현했으며, 인접한 노드를 먼저 탐색하였다. 역시 방문여부는 꼭 체크해주었다.