S2

DFS와 BFS (baekjoon 1260)

categories: 그래프 이론, 그래프 탐색, 너비 우선 탐색, 깊이 우선 탐색

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	128 MB	214420	79791	47417	36.157%

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 ≤ N ≤ 1,000), 간선의 개수 M(1 ≤ M ≤ 10,000), 탐색을 시작할 정점의 번호 V가 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.

문제 출처: BAE/JOON ONLINE JUDGE
https://www.acmicpc.net/problem/1260

GitHub
Woody K
Contact: woody35545@gmail.com
Github: https://github.com/woody35545
CNU, Computer Science and Engineering

S2

DFS와 BFS (baekjoon 1260)

categories: 그래프 이론, 그래프 탐색, 너비 우선 탐색, 깊이 우선 탐색

예제입력

```
4 5 1
1 2
1 3
1 4
2 4
3 4
```

예제출력

```
1 2 4 3
1 2 3 4
```

예제입력

```
5 5 3
5 4
5 2
1 2
3 4
3 1
```

예제출력

```
3 1 2 5 4
3 1 4 2 5
```

예제입력

```
1000 1 1000
999 1000
```

예제출력

```
1000 999
1000 999
```

Woody K
Contact: woody35545@gmail.com
Github: https://github.com/woody35545
CNU, Computer Science and Engineering

S2

DFS와 BFS (baekjoon 1260)

categories: 그래프 이론, 그래프 탐색, 너비 우선 탐색, 깊이 우선 탐색

Answer(1/2)

```python
def dfs(graph, start_node):
    stack, visited = [start_node], []

    while stack:
        cur_node = stack.pop()
        if cur_node in visited:
            continue

        visited.append(cur_node)

        temp = []
        for adj_node in graph[cur_node]:
            if adj_node not in visited:
                temp.append(adj_node)

        temp.sort(reverse=True)
        stack.extend(temp)

    return visited


def bfs(graph, startVertex):
    visited, queue = [], [startVertex]
    while queue:

        cur_node = queue.pop(0)  # left pop

        if cur_node not in visited:
            visited.append(cur_node)  # if current node's visited state is 'not visited', set 'visited'
            temp = graph[cur_node]
            temp.sort()
            queue.extend(temp)  # push current node's adjacent nodes to queue

    return visited

```


Woody K
Contact: woody35545@gmail.com
Github: https://github.com/woody35545
CNU, Computer Science and Engineering

S2

DFS와 BFS (baekjoon 1260)

categories: 그래프 이론, 그래프 탐색, 너비 우선 탐색, 깊이 우선 탐색

Answer(2/2)

```python
N, M, V = map(int, input().split(" "))
input_list = [input() for _ in range(M)]

graph = [[] for _ in range(N + 1)]  # there is no node named '0' but, add '0' for alignment
for i in range(len(input_list)):
    node1, node2 = map(int, input_list[i].split(" "))
    # Considering it's undirected graph
    graph[node1].append(node2)
    graph[node2].append(node1)


# solve
dfs_res_list = dfs(graph, V)
for i in range(len(dfs_res_list)):
    if i != (len(dfs_res_list) - 1):
        print(str(dfs_res_list[i]), end=" ")
    else:
        print(str(dfs_res_list[i]))

bfs_res_list = bfs(graph, V)
for i in range(len(bfs_res_list)):
    if i != (len(bfs_res_list) - 1):
        print(str(bfs_res_list[i]), end=" ")
    else:
        print(str(bfs_res_list[i]), end="")
```

GitHub

Woody K
Contact: woody35545@gmail.com
Github: https://github.com/woody35545
CNU, Computer Science and Engineering