1260 DFS와 BFS

2021년 4월 3일 토요일 오후 5:29

DFS와 BFS 분류

Silver II

난이도 제공: solved.ac

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	128 MB	126165	44830	25775	33.726%

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

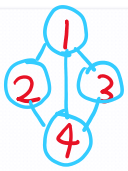
첫째 줄에 정점의 개수 N(1 ≤ N ≤ 1,000), 간선의 개수 M(1 ≤ M ≤ 10,000), 탐색을 시작할 정점의 번호 V가 주어진다. 다음 M개의 줄에는 간선이 연결하는 두 정점의 번호가 주어진다. 어떤 두 정점 사이에 여러 개의 간선이 있을 수 있다. 입력으로 주어지는 간선은 양방향이다.

출력

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

예제 입력 1 복사

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

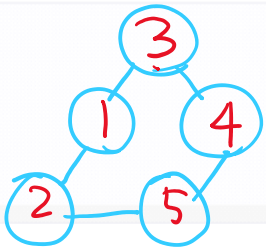


예제 출력 1 복사

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

예제 입력 2 복사

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



예제 출력 2 복사

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

예제 입력 3 복사

```
1000 1 1000
999 1000
```

예제 출력 3 복사

```
1000 999
1000 999
```

```python
input_data = list(map(int, input().split()))
# 정점
dot = input_data[0]
# 간선
line = input_data[1]
```

```
# 시작
start = input_data[2]
# 그래프 2차원 리스트로 구현
graph = []
# 빈 그래프를 생성
for i in range(dot+1):
    graph.append([]);
# 그래프 생성
# 딕셔너리 식으로 해당 자리에 input
for i in range(line):
    temp = list(map(int, input().split()))
    # graph.append([temp[0],temp[1]])
    graph[temp[0]].append(temp[1])
    graph[temp[1]].append(temp[0])
for i in range(len(graph)):
    graph[i].sort()
# dfs 함수
def dfs(graph, visit, visited):
    visited[visit] = True
    print(visit, end=' ')
    #현재 노드와 연결된 다른 노드를 재귀 방문
    for i in graph[visit]:
        if not visited[i]:
            dfs(graph, i, visited)

visited_dfs = [False] * (dot+1)
#bfs 함수
def bfs(graph, start):
    visit = list()
    queue = list()
    # 큐에 시작 노드를 넣어줌
    queue.append(start)
    while queue:
        # 맨 앞의 숫자를 뺀다
        node = queue.pop(0)
        if node not in visit:
            print(node, end=' ')
            visit.append(node)
            queue.extend(graph[node])


dfs(graph, start, visited_dfs)
print()
bfs(graph, start)
dfs(graph, start, visited)
```