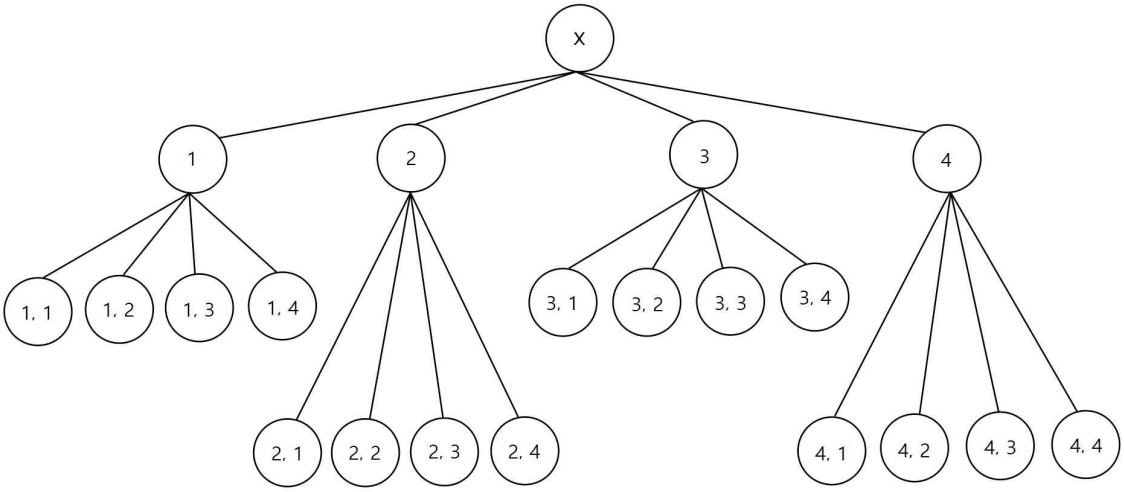
한 경로를 따라가다가 자신이 찾는 해가 나오지 않으면 왔던 길(상위 노드)로 되돌아가 해를 찾는 알고리즘
모든 경우의 수를 전부 고려해야 할 때 적합함

DFS에 적합함 단, 트리의 깊이가 무한대일 때 무한루프에 주의
BFS로 하면 큐나 스택의 크기가 너무 커질 수 있음



① 반복문을 사용하여 자신이 찾는 해의 조건에 맞으면 트리에 저장
(조건이 맞지 않으면 노드가 아예 만들어지지 않음)
② dfs 호출
③ dfs 탈출 조건 : 자신이 찾는 해를 찾았을 때
④ 해를 찾았으면 트리에서 다시 상위 노드로 올라감

①, ② : 노드 생성
③, ④ : 상위 노드로 올라가기

예시 : 15651 - N과 M (3)

```python
import sys

read = sys.stdin.readline

def dfs() :
    if len(s) == m :
        print(' '.join(map(str, s)))
        return

    for i in range(1, n+1) :
        s.append(i)
        dfs()
        s.pop()

n, m = map(int, read().split())
s = []
dfs()
```