

BOJ 17352.

여러분의 다리가 되어드리겠습니다!

소프트웨어학과 201921017 이지우

Step1 – 접근

문제

선린월드에는 N 개의 섬이 있다. 섬에는 $1, 2, \dots, N$ 의 번호가 하나씩 붙어 있다. 그 섬들을 $N - 1$ 개의 다리가 잇고 있으며, 어떤 두 섬 사이든 다리로 왕복할 수 있다.

어제까지는 그랬다.

"왜 다리가 $N - 1$ 개밖에 없냐, 통행하기 불편하다"며 선린월드에 불만을 갖던 옥제가 다리 하나를 무너뜨렸다! 안 그래도 불편한 통행이 더 불편해졌다. 서로 왕복할 수 없는 섬들이 생겼기 때문이다. 일단 급한 대로 정부는 선린월드의 건축가를 고용해, 서로 다른 두 섬을 다리로 이어서 다시 어떤 두 섬 사이든 왕복할 수 있게 하라는 지시를 내렸다.

그런데 그 건축가가 당신이다! 안 그래도 천하제일 코딩대회에 참가하느라 바쁘는데...

입력

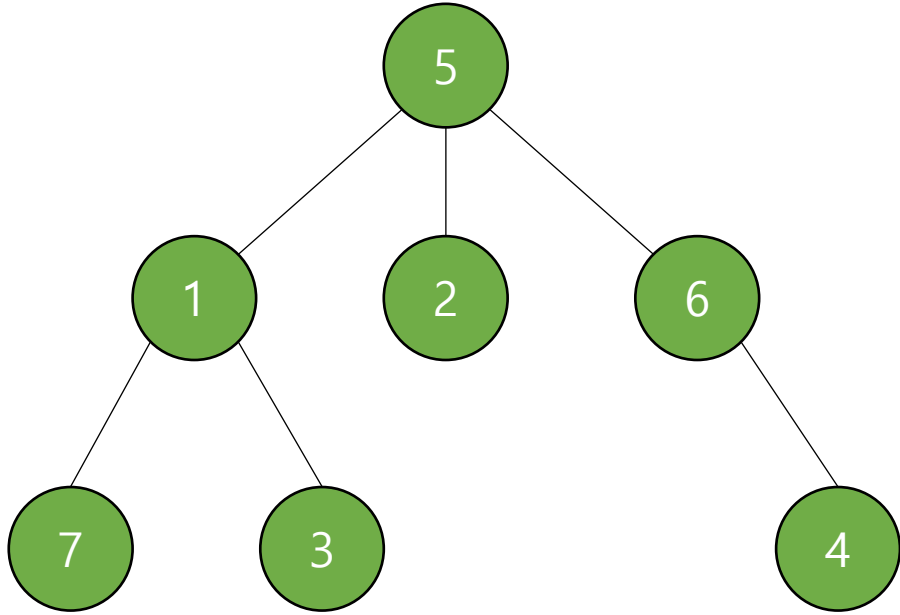
첫 줄에 정수 N 이 주어진다. ($2 \leq N \leq 300,000$)

그 다음 $N - 2$ 개의 줄에는 옥제가 무너뜨리지 않은 다리들이 잇는 두 섬의 번호가 주어진다.

출력

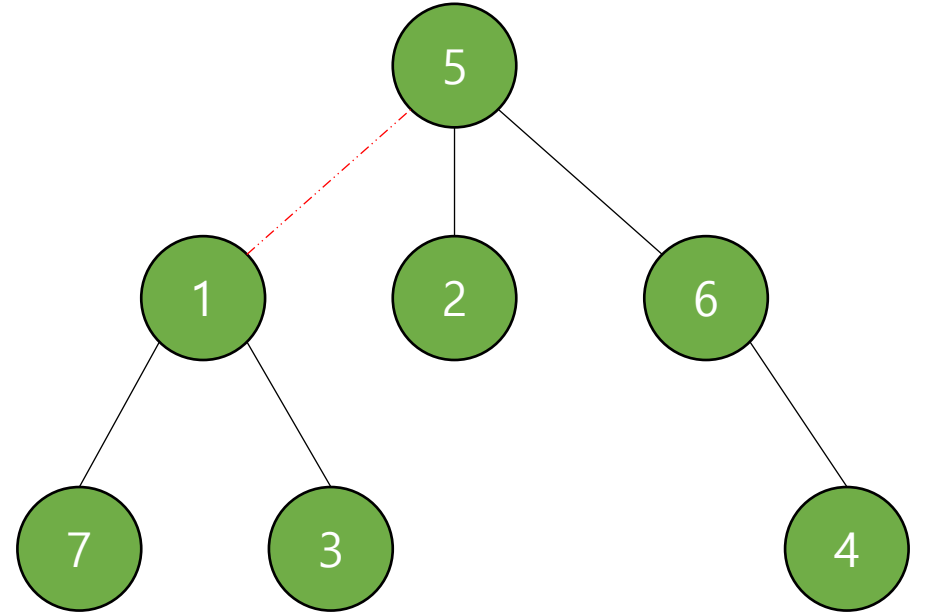
다리로 이을 두 섬의 번호를 출력한다. 여러 가지 방법이 있을 경우 그 중 아무거나 한 방법만 출력한다.

Step1 - 접근



[어제의 섬: $n-1$ 개의 다리(간선)로 연결된 트리]

>>



[오늘의 섬: $n-1$ 개의 다리(간선)로 연결된 그래프]

Step1 - 접근

다리로 이을 두 개의 섬 출력(아무거나 상관 X)

→ K개에 속한 임의의 섬에서 DFS 진행

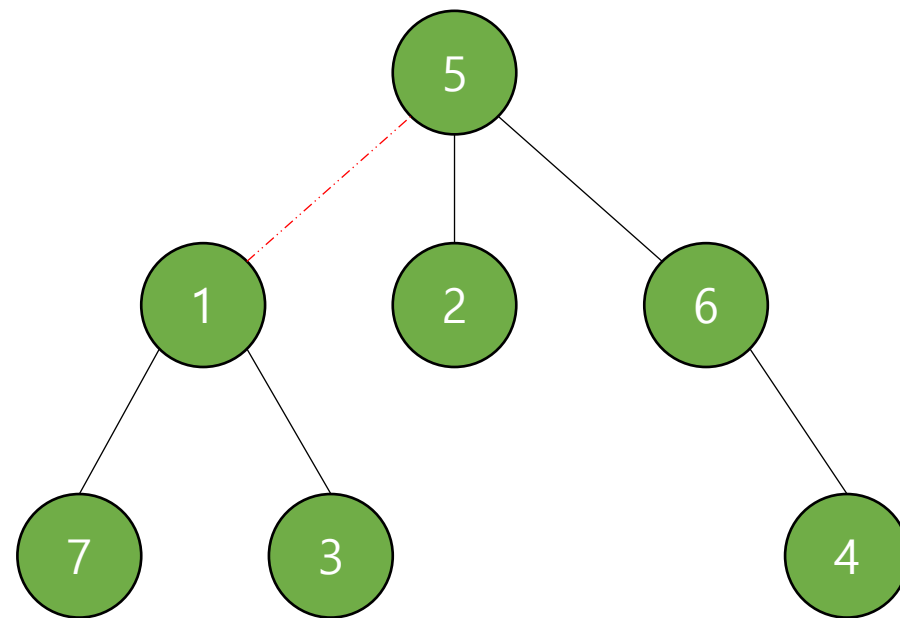
→ 다리를 통해 갈 수 있는 모든 섬 탐색

→ DFS 종료 후 아직 방문하지 않은 임의의 섬 하나 찾기

→ 그 섬은 N-K개에 속한 섬임!

→ 두 임의의 섬을 연결해주는 다리를 만들면!

→ N-1개의 다리가 다시 생김~



Step2 - 구현

```
from sys import stdin
from collections import defaultdict as dd

def dfs(start):

    s = list([start])

    while s:
        curr = s.pop()
        visited[curr] = True

        for adj in adj_list[curr]:
            if not visited[adj]:
                s.append(adj)
```

DFS 구현

```
n = int(stdin.readline())
adj_list = dd(list)

for _ in range(n-2):
    v1, v2 = map(int, stdin.readline().split())

    adj_list[v1].append(v2)
    adj_list[v2].append(v1)

visited = [False] * (n+1)

dfs(1)

for i in range(1, n+1):
    if not visited[i]:
        print(i)
```

인접 리스트 생성

dfs(1) 1. 임의의 한 섬(1)에서 DFS진행

2. 떨어져 있는 섬 발견!

3. 1, 2의 섬을 서로 이어줌

break

THX~