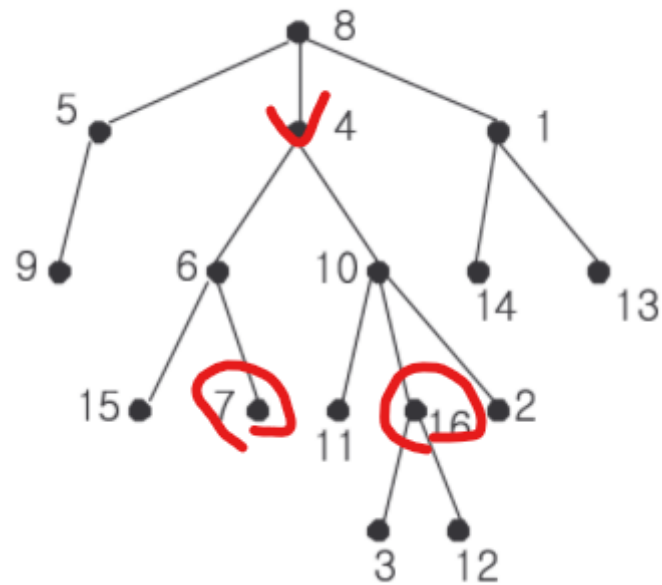


트리 3

🕒 Created	@December 9, 2021 6:13 PM
🏷 Tags	
📁 난이도	

가장 가까운 공통 조상 3584



공통 부모 노드를 찾자.

풀이

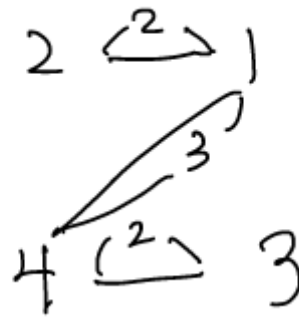
1. 부모 노드를 기억하고 있는 테이블 유지
2. 시작 노드들의 부모 노드를 추적하며 visit 상태로 변경
3. 두번째 추적 때 처음으로 만나는 visit 노드 를 출력

시간 복잡도

$O(\text{노드의 최고 레벨} * 2)$

노드 사이의 거리 1240

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



트리에서 두 노드 사이의 경로는 유일하다. BFS도 사용 가능

풀이

1. 가중치 양방향 인접 리스트 유지
2. 시작지에서 도착지 까지 접근

시간 복잡도

$O(N^2)$

트리와 쿼리 15681

트리가 주어진다.

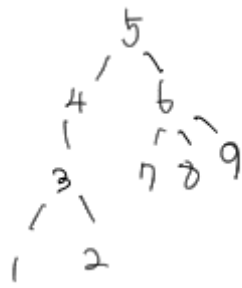
예제 입력 1 복사

```
9 5 3
1 3
4 3
5 4
5 6
6 7
2 3
9 6
6 8
5
4
8
```

예제 출력 1 복사

```
9
4
1
```

위의 입력은 아래 트리와 같다.



가장 마지막에 주어진 쿼리를 루트로 한 서브트리의 정점 수를 세어야한다.

방법 1. 쿼리마다 서브 트리의 정점 수를 센다

양방향 인접 리스트를 저장하기 때문에 어디가 자식인지 알 수가 없다.

방법 2. DFS로 자식 노드 숫자를 세기

```
private static void dfs(int pre, int cur) {  
    childTable[cur] = 1; // 리프 노드는 1 저장 후 끝난다.  
    for (int next : tree[cur]) {  
        if (next == pre) continue;  
        dfs(cur, next); // 리프 노드까지 들어가기  
        childTable[cur] += childTable[next]; // next의 자식 노드 숫자 더하기  
    }  
}
```