

# 11725번: 트리의 부모 찾기

11725번 제출 맞은 사람 쏫코딩 풀이 풀이 작성 풀이 요청 재채점/수정 문제 추천

채점 현황 내 소스 강의▼ 질문 검색 질문 작성

## 트리의 부모 찾기

성공 풀이

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	256 MB	3273	1344	1027	43.609%

## 문제

루트 없는 트리가 주어진다. 이 때, 트리의 루트를 1이라고 정했을 때, 각 노드의 부모를 구하는 프로그램을 작성하시오.

입력

첫째 줄에 노드의 개수  $N$  ( $1 \leq N \leq 100,000$ )이 주어진다. 둘째 줄부터  $N-1$ 개의 줄에 트리 상에서 연결된 두 정점이 주어진다.

## 출력

첫째 줄부터  $N-1$ 개의 줄에 각 노드의 부모 노드 번호를 2번 노드부터 순서대로 출력한다.

문제 유형 : 그래프 , 탐색 (BFS)

접근 방법 : 트리의 부모 노드를 찾는 문제이다 .부모를 찾으려면 인접한 노드들 중 어느 쪽이 루트와 가까운지 알아야한다. 이러한 관계를 알기위해 일단 BFS를 돌려서 루트로 부터의 거리를 기록 했다. 그후 모든 노드를 방문하며 인접 노드중 최소 값을 가진 곳을 부모 노드라고 파악해서 출력했다.

- 1) 그래프 입력 받기
- 2) BFS 돌려 root 로부터의 거리를 측정
- 3) 노드를 순차 방문하여 부모노드 탐색

```
import java.io.BufferedReader;
```

```

import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.Queue;
class pair{
    int x;
    ArrayList <Integer> y;
    pair (int x, ArrayList<Integer> y){
        this.x=x;
        this.y=y;
    }
}

}

public class Main {
    public static BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
    public static ArrayList<Integer> cnj[];
    public static int [] roots ;
    public static Queue<pair> q = new LinkedList<>();
    public static int N;
    public static void main(String args[]) throws Exception{
        getInput();
        BFS(0);
//        for (int i : roots) { //          System.out.print(i+" "); //      } System.out.println();
        for (int i = 1; i < N; i++) {
            int tmp = roots[i];
            for (int j : cnj[i]) {
                if (tmp > roots[j]) System.out.println((j+1));
            }
        }
    }

    public static void BFS(int root) {
        q.add(new pair(1, cnj[root]));
        roots[root] = 1;
        while (!q.isEmpty()) {

            pair targets = q.poll();
            int depth = targets.x+1;
            for (int r : targets.y) {
                if (roots[r] == 0) { q.add(new pair(depth, cnj[r]));
                    roots[r] = depth ;
                }
            }
        }
    }

    public static void getInput() throws Exception {
        N = Integer.parseInt(in.readLine());
        cnj = new ArrayList[N];
        for (int i = 0; i < N; i++) {
            cnj[i] = new ArrayList<Integer>();
        }
        roots = new int[N];
        for (int i = 0; i < N-1; i++) {
            int a, b;
            String line[] = in.readLine().split(" ");
            a = Integer.parseInt(line[0])-1; b = Integer.parseInt(line[1])-1;
            cnj[a].add(b);
            cnj[b].add(a);
        }
    }
}

```

```
}  
}
```

## 예제 입력

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

## 예제 출력

```
4  
6  
1  
3  
1  
4
```

## 예제 입력 2

```
12  
1 2  
1 3  
2 4  
3 5  
3 6  
4 7  
4 8  
5 9  
5 10  
6 11  
6 12
```

## 예제 출력 2

```
1
1
2
3
3
4
4
5
5
6
6
```

## 힌트

## 출처

- 문제를 만든 사람: baekjoon