

## Problem I. Query On A Tree 17

Input file:            standard input  
Output file:          standard output  
Time limit:           2 seconds  
Memory limit:        1024 megabytes

On *Baekjoon Online Judge*, there are a series of problems related to processing queries on a tree. We KAISTians are proud to offer the seventeenth edition in this series.

You are given a tree with  $N$  vertices. Each vertex is numbered from 1 to  $N$ . The tree is rooted at vertex 1.

People can live in each vertex. Let  $A[i]$  be the number of people living in vertex  $i$ . Initially,  $A[i] = 0$  for all  $1 \leq i \leq N$ .

Write a program that processes the  $Q$  following queries:

- 1  $u$ : Add 1 to  $A[i]$  for all vertices  $i$  in the subtree rooted at vertex  $u$ .
- 2  $u$   $v$ : Add 1 to  $A[i]$  for all vertices  $i$  on the unique shortest path between the two vertices  $u$  and  $v$ . Note that  $u$  and  $v$  might be equal.

After each query, print the vertex  $x$  that minimizes the quantity  $\sum_{y=1}^N A[y] \times dist(x, y)$ , where  $dist(x, y)$  is the number of edges on the path between  $x$  and  $y$ . If there is more than one such vertex, print the vertex with minimum distance from the root (vertex 1). It can be proven that such vertex is unique. In other words, we should find a vertex that minimizes the total distance needed for everyone to gather.

### Input

The first line contains the single integer  $N$  ( $2 \leq N \leq 100\,000$ ).

The next  $N - 1$  lines describe all edges of the tree. Each line contains two space-separated integers  $u, v$  denoting that there is an edge connecting vertices  $u$  and  $v$ . ( $1 \leq u, v \leq N, u \neq v$ )

The next line contains the single integer  $Q$  ( $1 \leq Q \leq 100\,000$ ).

The next  $Q$  lines contains several integers in one of the following forms:

- 1  $u$  ( $1 \leq u \leq N$ )
- 2  $u$   $v$  ( $1 \leq u, v \leq N$ )

### Output

Print  $Q$  lines, denoting the answer after each update.

### Example

standard input	standard output
7	2
1 6	7
1 7	7
7 3	1
3 2	
7 5	
5 4	
4	
1 2	
1 4	
1 6	
2 6 7	