

IOI Training Camp 2013 – Test 4, 4 May, 2013

Ancestors Sum

You have just learnt that the Binary Indexed Tree is useful in dynamically keeping track of prefix sums. This problem prompts you to generalise this data-structure even further.

You are given a tree with N whose vertices are numbered $\{0, 1, \dots, N - 1\}$, where the root is numbered 0. Each vertex stores an integer value. Initially, the values in all the vertices are 0. The tree supports one *update* operation: Given a vertex v and a value inc , the operation increments the value at vertex v by inc . The tree also supports one *query* operation: Given a leaf node l , the query returns the sum of all the values along the path from the leaf to the root (inclusive of the values at the leaf and the root).

This problem requires you to design a generalisation of the Binary Indexed Tree data structure (call it the *Ancestor Sum* data structure) to solve the following: given a sequence of Q tree operations (could be a combination of *update* and *query* operations), you are required to output the values for all the *query* operations.

Input format

- The first line consists of the single integer N
- The next $N - 1$ lines contain space separated pairs u and v denoting an edge of the tree is between vertex u and vertex v .
- The next line contains the number of tree operations Q .
- Finally, the subsequent Q lines contains tree operations. The tree operation could have one of the following forms:
 - The character ' A ' followed by two space separated integers v and inc . This corresponds to the *update* operation for incrementing the value at vertex v by inc .
 - The character ' S ' followed by an integer l . This corresponds to the *query* operation for the sum of values in the vertices along the path from the leaf l to the root.

Output format

For each ' S ' l query, output the required sum.

Test data

In all the subtasks, $inc \leq 1,000,000$.

- Subtask 1 (30 marks) : $N \leq 1000$, $Q \leq 1000$.
- Subtask 2 (70 marks) : $N \leq 100,000$, $Q \leq 100,000$

Sample input

6
0 1
1 3
1 4
0 2
2 5
19
A 1 5
S 3
S 4
S 5
A 3 6
S 3
S 4
S 5
A 0 1
S 3
S 4
S 5
A 1 1
S 3
S 4
S 5
A 5 9
S 5
S 3

Sample output

5
5
0
11
5
0
12
6
1
13
7
1
10
13

Limits

- *Memory limit* : 128 MB
- *Time limit* : 4s