Given a rooted tree of $N$ nodes, where each node is uniquely numbered in between *[1..N]*. The node 1 is the root of the tree. Each node has an integer value which is initially 0.

You need to perform the following two kinds of queries on the tree:

- *add t value*: Add value to all nodes in subtree rooted at *t*
- *max a b*: Report maximum value on the path from *a* to *b*

**Input Format**

First line contains *N*, number of nodes in the tree. Next *N-1* lines contain two space separated integers *x* and *y* which denote that there is an edge between node *x* and node *y*.
Next line contains *Q*, the number of queries to process.
Next *Q* lines follow with either *add* or *max* query per line.

**Constraints**

$1 \leq N \leq 10^5$
$1 \leq Q \leq 10^5$
$1 \leq t, a, b, x, y \leq N$
$x \neq y$
$-10^4 \leq value \leq 10^4$

**Output Format**

For each *max* query output the answer in a separate line.

**Sample Input**

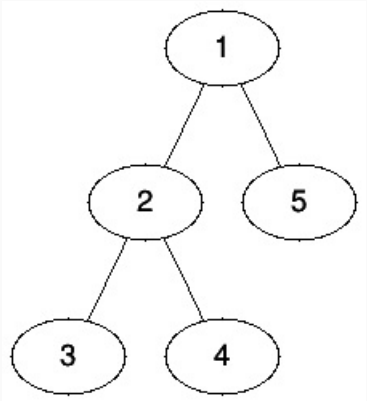
```
5
1 2
2 3
2 4
5 1
6
add 4 30
add 5 20
max 4 5
add 2 -20
max 4 5
max 3 4
```

**Sample Output**

```
30
20
10
```

**Explanation**

In the test case we have the following tree:




Initially all node values are zero.

Queries are performed in the following way:

**add 4 30** // add 30 to node 4
**add 5 20** // add 20 to node 5
**max 4 5** // maximum of nodes 4,2,1,5 is 30
**add 2 -20** // subtract 20 from nodes 2,3,4
**max 4 5** // maximum of nodes 4,2,1,5 is 20
**max 3 4** // maximum of nodes 3,2,4 is 10