Heavy-light decomposition P53018_en

Dotzè Concurs de Programació de la UPC - Semifinal (2014-07-02)

Let T be a rooted tree with n nodes. For every node u, let $S(u)$ denote the size (the number of nodes) of the subtree rooted at u, including u itself. Let p_u denote the parent of u.

An edge $\{u, p_u\}$ is called *heavy* if $2S(u) > S(p_u)$, and *light* otherwise. Note that at most one child of every node can be connected by a heavy edge. It is easy to see that the number of light edges in the path from any node to the root of T is at most $O(\log n)$. Moreover, the connected components of the subgraph that only uses the heavy edges are just paths and single nodes.

Now consider the classic Lowest Common Ancestor Problem (given two nodes, find the closest node to them that is an ancestor of both). The following simple code returns the LCA of two given nodes in cost $O(\log n)$.

```
int lca(int u, int v) {
  while (head[u] != head[v]) {
    if (depth[head[u]] > depth[head[v]]) u = parent[head[u]];
    else v = parent[head[v]];
  }
  return depth[u] < depth[v] ? u : v;
}
```

where `depth` and `parent` mean exactly what their names suggest and `head[u]` is the node closest to the root that belongs to the same heavy path as u (maybe u itself).

With all the information above, perhaps you are ready to solve the following problem. Given an unrooted tree with costs at the edges, perform two types of queries:

- `c` u v x: Change to x the cost of the edge $\{u, v\}$.
- `q` u v: Print the sum of the costs in the path from u to v.

Input

Input consists of several cases. Every case begins with n and the number of queries m. Follow $n - 1$ triples u v x, meaning that u and v are connected by an edge of cost x. Follow m queries. Assume $2 \leq n \leq 10^4$, $1 \leq m \leq 10^4$, that the nodes are numbered from 0, and that all costs are integers between 1 and 10^9.

Output

Print the answer to each query of the second type, and a blank line after every case.

Hint

Beware of overflows.

Sample input

```
12 10
0 1 4
0 4 1
1 2 7
1 5 2
1 7 3
2 3 5
2 6 2
6 11 3
7 8 2
7 9 1
7 10 4
q 2 4
q 1 7
q 0 4
q 3 3
q 8 1
q 4 2
c 1 7 10
c 6 11 1
q 11 0
q 8 4

4 5
0 1 1
1 2 1
2 3 1
q 0 1
q 0 2
q 0 3
c 1 2 10
q 0 3
```

Sample output

```
12
3
1
0
5
12
14
17

1
2
3
12
```

Problem information

Author : Alex Alvarez
Generation : 2024-05-02 19:12:23