

---

# Yet Another Tree Problem

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            5 seconds  
Memory limit:         512 megabytes

Given a tree with  $n$  vertices (a tree is an undirected connected simple graph without cycles). Each edge has an integer weight. Some vertices of the tree are marked. You need to implement a program that performs the following operations on this tree:

1. Mark the vertex. It is guaranteed that before this operation it was not marked.
2. Unmark the vertex. It is guaranteed that before this operation it was marked.
3. Change the weight of the edge.

Before performing all operations and after each operation, your program should output the maximum number  $x$  such that there is a simple path with **at least two** marked vertices on it, with total weight equal to  $x$ . If there are no two marked vertices in the tree, print «BAD».

## Input

The first line contains two integers  $n$  and  $q$  ( $1 \leq n, q \leq 150\,000$ ), the number of vertices and the number of operations, respectively.

The next line contains  $n$  numbers, describing whether the vertex is marked in the beginning: 1 means that the vertex is marked, 0 means that that the vertex is not marked.

The next  $n - 1$  lines describe the edges of the tree. For all  $i$  from 2 to  $n$ , there are two numbers  $p_i$  ( $1 \leq p_i < i$ ) and  $w_i$  ( $-10^9 \leq w_i \leq 10^9$ ), which means that there is an edge in the tree between the vertices  $i$  and  $p_i$  of weight  $w_i$ .

The next  $q$  lines describe the operations. Each operation has one of the following forms:

- 1  $x$  ( $1 \leq x \leq n$ ): mark vertex  $x$ ,
- 2  $x$  ( $1 \leq x \leq n$ ): unmark vertex  $x$ ,
- 3  $x w$  ( $2 \leq x \leq n, -10^9 \leq w \leq 10^9$ ): make the weight of the edge between the vertices of  $p_x$  and  $x$  equal to  $w$ .

## Output

Output  $q + 1$  lines. In the first line print the answer before performing the operations. On each next line print the answer after performing the next operation.

## Scoring

By  $D$  we denote the maximum number of edges connected to a single vertex, and by  $H$  the maximum number of vertices on a simple path from vertex 1 to some other vertex.

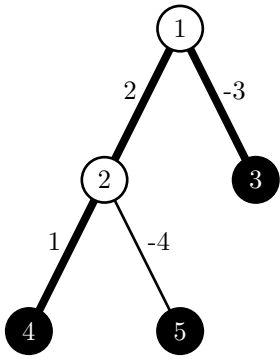
Subtask	Score	Constraints		
		$n$	$q$	Additional
1	3	$n \leq 20$	$q \leq 20$	—
2	3	$n \leq 500$	$q \leq 500$	—
3	3	$n \leq 100$	$q \leq 5000$	—
4	5	$n \leq 5000$	$q \leq 5000$	—
5	7	$n \leq 10^5$	$q \leq 10^5$	$D \leq 30, H \leq 20$
6	20	$n \leq 10^5$	$q \leq 10^5$	$H \leq 20$
7	10	$n \leq 10^5$	$q \leq 10^5$	$p_v = v - 1$ for all $v$
8	17	$n \leq 10^5$	$q \leq 10^5$	$w_i \leq 0, w \leq 0$ for all operations
9	22	$n \leq 50\,000$	$q \leq 50\,000$	—
10	5	$n \leq 10^5$	$q \leq 10^5$	—
11	5	$n \leq 150\,000$	$q \leq 150\,000$	—

## Example

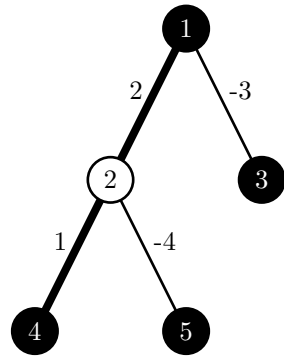
standard input	standard output
5 7	0
0 0 1 1 1	3
1 2	0
1 -3	8
2 1	3
2 -4	BAD
1 1	BAD
2 4	6
3 3 5	
2 3	
2 1	
3 5 -1	
1 1	

## Note

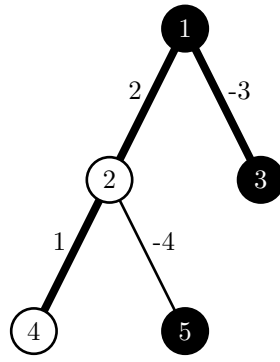
Pictures for the example. The bold path is the path of weight  $x$  with at least two marked vertices.



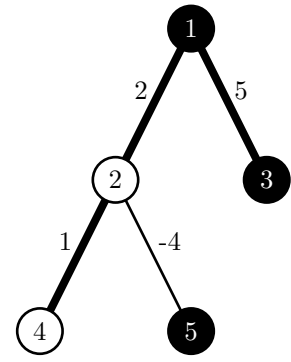
before operations



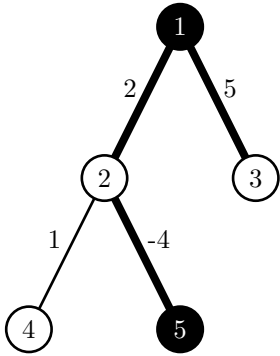
after operation 1



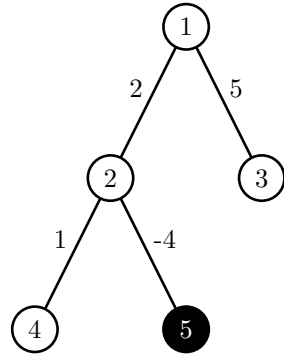
after operation 2



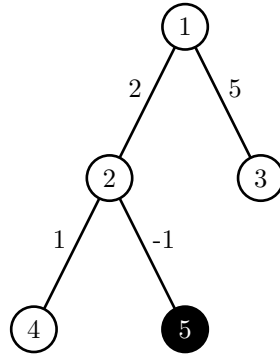
after operation 3



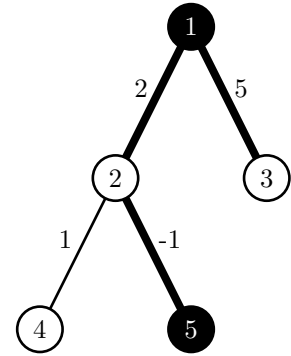
after operation 4



after operation 5



after operation 6



after operation 7