Query on Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	512 megabytes

Given a rooted tree with n nodes, where the root is node $r = 1$, each node has an associated weight a_i.

The distance between two nodes in the tree is defined as the number of edges on the shortest path between them.

The subtree of a node x is defined as the set of all nodes y such that the shortest path from y to the root r passes through x.

You need to support the following three types of operations:

1. Given x, k, and v, for each node y at a distance exactly k from x, update a_y to $a_y + v$, then output the maximum of a_y.

2. Given x, k, and v, for each node y at a distance of at most k from x, update a_y to $a_y + v$, then output the maximum of a_y.

3. Given x and v, update a_y to $a_y + v$ for every node y in the subtree of x, then output the maximum of a_y.

Input

The first line contains an integer T ($1 \le T \le 10^4$), representing the number of test cases.

For each test case, the first line contains two integers n and q ($1 \le n, q \le 2 \times 10^5$), representing the number of nodes in the tree and the number of operations.

The second line contains n integers $a_1, a_2, \cdots, a_n$ ($-10^{14} \le a_i \le 10^{14}$), representing the weight of each node.

The next $n - 1$ lines each contain two integers x and y ($1 \le x, y \le n$, $x \ne y$), representing an edge in the tree.

The next q lines describe the operations. Each line contains three or four integers o, x, v or o, x, k, v ($o \in \{1, 2, 3\}$, $1 \le x \le n$, $0 \le k < 10$, $-10^9 \le v \le 10^9$), representing the operation type and its parameters.

It is guaranteed that the sum of n over all test cases does not exceed 5×10^5, and the sum of q over all test cases does not exceed 2×10^5.

Output

For each test case, output q lines, each containing an integer representing the result of the corresponding operation. If no valid node y exists, output `GG`.

Example

standard input	standard output
1 5 5 1 2 1 3 2 1 2 2 3 2 4 4 5 2 2 1 0 1 2 1 3 3 4 -5 2 5 2 3 3 2 -1	3 6 1 5 4

Note

The initial node weights are $a = [1, 2, 1, 3, 2]$.

After the 1st operation, the weights remain $a = [1, 2, 1, 3, 2]$.

After the 2nd operation, the weights become $a = [4, 2, 4, 6, 2]$.

After the 3rd operation, the weights become $a = [4, 2, 4, 1, -3]$.

After the 4th operation, the weights become $a = [4, 5, 4, 4, 0]$.

After the 5th operation, the weights become $a = [4, 4, 3, 3, -1]$.