Problem I. Danger

Input file:	`danger.in`
Output file:	`danger.out`
Time limit:	5 seconds
Memory limit:	256 megabytes

You are given an undirected tree consisting of N nodes with weights assigned to edges. You need to perform queries of two types:

1. Add a given value a_i to all edges on the path from node v_i to node u_i. There will be no more than 500 queries of this type in the input.
2. Compute the sum of weights of all edges for which both endpoints are at distance no more than k_i from node v_i. The distance between two nodes is the number of edges on the path between them.

You should perform these queries in the given order and for each query of second type, print the computed value.

Input

The first line of input contains two integers N and Q ($1 \le N \le 1000$, $0 \le Q \le 500\,000$) which are the number of nodes in the tree and the number of queries, respectively. The next $N - 1$ lines contain the description of the tree: i-th of these lines contains three integers v_i, u_i and w_i ($1 \le v_i, u_i \le N$, $1 \le w_i \le 100\,000$) which are the number of nodes this edge connects and its weight, respectively. It is guaranteed that the given graph is a tree.

The following Q lines contain the description of queries. On each of these lines, the first integer t_i is the type of the query ($1 \le t_i \le 2$). If $t_i = 1$, it is followed by three integers v_i, u_i and a_i ($1 \le v_i, u_i \le N$, $1 \le a_i \le 100\,000$) and means you need to add a_i to all edges on the path from v_i to u_i. If $t_i = 2$, it is followed by two integers v_i and k_i ($1 \le v_i \le N$, $1 \le k_i \le N$) and means you need to find the sum of weights of all edges for which both endpoints are at distance at most k_i form node v_i. It is guaranteed that the number of queries of the first type will not be greater than 500.

Output

For each query of the second type, print exactly one integer on a separate line: the answer to the query.

Examples

danger.in	danger.out
5 5 1 2 1 2 3 2 3 4 3 3 5 1 2 2 1 2 4 2 1 1 4 2 2 1 2 2 3 3	3 6 7 13