Intersection of Paths

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

There is a tree with n vertices and $(n-1)$ edges, where the i-th edge connects vertices u_i and v_i, and has a weight of w_i.

Your task is to process q queries. The i-th query can be described as three integers a_i, b_i and k_i. This query will temporarily change the weight of the a_i-th edge to b_i. After that you should choose $2k_i$ distinct vertices $s_1, s_2, \cdots, s_{k_i}, e_1, e_2, \cdots, e_{k_i}$ and consider the k_i simple paths on the tree, where the p-th path starts from vertex s_p and ends at vertex e_p. We say an edge is good, if it is contained in all k_i paths. Maximize the total weights of good edges.

Note again that the change in the weight of each query is temporary. After each query you should change back the weight.

Input

There is only one test case in each test file.

The first line contains two integers n and q ($2 \le n \le 5 \times 10^5$, $1 \le q \le 5 \times 10^5$) indicating the number of vertices and the number of queries.

For the following $(n-1)$ lines, the i-th line contains three integers u_i, v_i and w_i ($1 \le u_i, v_i \le n$, $1 \le w_i \le 10^9$) indicating that the i-th edge connects vertices u_i and v_i, and has a weight of w_i.

For the following q lines, the i-th line contains three integers a_i, b_i and k_i ($1 \le a_i \le n-1$, $1 \le b_i \le 10^9$, $1 \le k_i \le \lfloor \frac{n}{2} \rfloor$) indicating the i-th query.

Output

For each query output one line containing one integer indicating the answer.

Example

standard input	standard output
7 3 1 2 20 2 3 10 2 4 40 4 6 10 1 5 30 5 7 10 2 100 1 5 50 2 2 100 3	160 110 20

Note

For the first query, choose $s_1 = 3$ and $e_1 = 7$.

For the second query, choose $s_1 = 4$, $s_2 = 6$, $e_1 = 7$ and $e_2 = 5$.

For the third query, choose $s_1 = 3$, $s_2 = 4$, $s_3 = 6$, $e_1 = 5$, $e_2 = 1$ and $e_3 = 7$.