Crawling on a Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	1024 megabytes

Yet, confronted with what seems like an endless expanse of future possibilities, I find myself inclined towards Calvino's notion in "The Baron in the Trees" - a life lived amidst the branches appears preferable to prematurely taking flight.

— *Living in a Tree*

There is a tree with n vertices, labeled by $1, 2, \ldots, n$. At the 1-st vertex, there are m turtles. Each turtle can crawl along the bidirectional edges of the tree to reach other vertices. Turtles are very heavy, so for the i-th edge, after the k_i-th time that a turtle passes it, the edge will be broken such that turtles can't pass again. Note that multiple turtles can pass the same edge at the same moment. Assuming there are cnt turtles passing the same edge at the same moment, you should count cnt times for the edge. Of course, $cnt > k_i$ is not allowed.

Your task is to command the movement of these m turtles, such that for the i-th ($2 \le i \le n$) vertex, it will be visited by at least c_i turtles. Note that if a turtle visits a vertex for multiple times, it will be counted only once. Please find a movement to minimize the total distance of all turtles, or determine it is impossible.

Input

The first line of the input contains two integers n and M ($2 \le n \le 10^4, 1 \le M \le 10^4$), denoting the number of vertices and the upper bound of m.

Each of the next $(n-1)$ lines contains four integers u_i, v_i, l_i and k_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$, $1 \le l_i, k_i \le 10^9$), denoting a bidirectional edge between the u_i-th vertex and the v_i-th vertex, whose length is l_i, and it will be broken after the k_i-th time that a turtle passes it. It is guaranteed that the roads form a tree.

The next line contains $(n-1)$ integers $c_2, c_3, \ldots, c_n$ ($1 \le c_i \le M$), denoting the minimum number of turtles that each vertex will be visited by.

Output

Print M lines, the i-th ($1 \le i \le M$) line containing an integer, denoting the minimum total distance when $m = i$. If it is impossible to find a feasible movement, please print "`-1`" instead.

Examples

standard input	standard output
4 2 1 2 3 2 2 3 2 1 2 4 5 1 1 1 1	-1 13
4 2 1 2 3 2 2 3 2 1 2 4 5 1 2 2 2	-1 -1

Note

In the first example, when $m = 1$, it is impossible to let one turtle reach both vertex 3 and vertex 4. When $m = 2$, one of the possible solutions that minimizes the total distance is to let both turtles move from vertex 1 to vertex 2, then let the first turtle move to vertex 3, and let the second turtle move to vertex 4. The total distance traveled by the two turtles is $(3 + 2) + (3 + 5) = 13$.