Tree Fun

Time limit: *1000 ms*
Memory limit: *256 MB*

You are given a tree with N nodes, each node has a score which is initially 0.

The input contains the structure of the tree as well as a list of M operations. Each operation identifies a pair of nodes (A, B) and a value K. With each operation the value of each node in the path between A an B is increased by K, including the starting and ending nodes.

After performing all the operations in the input, print the maximum score in the tree.

Standard Input

The first line of the input contains two integers: the number of nodes N, and the number of operations M.

The next $N - 1$ lines contain 2 integers U_i, V_i denoting an edge between U_i and V_i.

The following M lines contain three integers A_i, B_i, K_i representing the starting node, ending node, and score to add along the path.
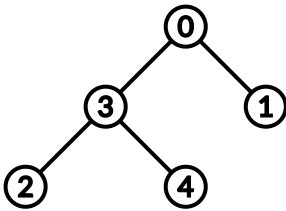
Standard Output

A single integer representing the maximum score in the tree.

Constraints and notes

- $1 \le N, M \le 10^5$
- $0 \le U_i, V_i, A_i, B_i < N$
- $1 \le K_i \le 1\,000$

Input	Output	Explanation

Input	Output	Explanation
5 2 0 1 0 3 2 3 3 4 4 1 4 2 4 3	7	$N = 5, M = 2$ This test case consists of a tree with 5 nodes in which we should do 2 operations.

The list of connected nodes forms a tree like this:

4 1 4 - updates the score from node 4 to node 1 by a value of 4

Path from 4 to 1 (4->3->0->1)

2 4 3 - updates the score from node 2 to node 4 by a value of 3

Path from 2 to 4 (2->3->4)

The maximum score is 7 (value of node 4 and 3)