Teleporter

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 megabytes

A country has n cities, with $n-1$ roads connecting them. The i-th road connects city u_i and v_i, costing w_i units of time to go through the road. Every city is accessible from another city via these roads, forming a tree. Since technology is very developed in the country, there is a teleporter in each city. There are m teleport paths between these teleporters. The i-th path connects the teleporter in city p_i and q_i. If a person is in city u, he can travel to city v without costing any time if there is a teleport path directly connecting the teleporters in these two cities. As teleporting is very expensive, people can only use the teleporter no more than k times during their trip.

Small N is the minister of transportation of the country. He wants to assess the efficiency of this transportation network. Precisely speaking, he wants to calculate $\sum_{u=1}^{n} d(u,k)$ for each $k = 0, 1, \cdots, n$ where $d(u,k)$ is the minimum time cost to travel from city u to 1 using teleporters no more than k times. As the calculation is too complex for him, he wants you to design a program to calculate for him.

Input

The first line contains two integers n and m ($1 \le n \le 5000, 0 \le m \le 10000$), indicating the number of cities and teleport paths.

The following $n-1$ lines contain three integers each; 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 road i connects city u_i and v_i, costing w_i units of time to go through the road.

The following m lines contain two integers each; the i-th line contains two integers p_i and q_i ($1 \le p_i, q_i \le n$), indicating teleport path i connects the teleporter in city p_i and q_i.

Output

Your output should contain $n+1$ lines; the i-th line should contain one integer indicating the result when $k = i - 1$.

Example

standard input	standard output
5 4 2 5 6 1 2 7 1 3 6 1 4 4 4 5 3 1 3 5 1 2	30 8 4 0 0 0