And Queries (`andqueries`)

You are given a weighted tree with N nodes numbered from 0 to $N-1$.

Let $cost(u, v)$ denote the bitwise `AND` of all edge weights on the shortest path between nodes u and v.

The cost of the tree is the sum of $cost(u, v)$ across all $0 \le u < v < N$. More formally, the cost of the tree is:

$$\sum_{u=0}^{N-2} \sum_{v=u+1}^{N-1} cost(u, v)$$

You have to process Q queries described by a triplet of integers U_j, V_j, X_j: the weight of the edge connecting nodes U_j and V_j changes to X_j.

Print the cost of the tree before the first query and following each query.

> ☞ Among the attachments of this task you may find a template file `andqueries.*` with a sample incomplete implementation.

Input

The first line of the input contains a single integer N, the number of nodes in the tree.

Each of the next $N-1$ lines contains 3 integers A_i, B_i and W_i, representing an edge between nodes A_i and B_i with weight W_i.

The next line contains a single integer Q, the number of queries.

Each of the next Q lines contains 3 integers U_j, V_j and X_j, describing a query.

Output

Print $Q + 1$ integers, the cost of the tree before the first query, and after each query.

Constraints

- $2 \le N \le 100\,000$.
- $0 \le A_i, B_i < N$ for each $i = 0 \dots N - 2$.
- The edges form a tree graph.
- $0 \le W_i < 2^{30}$ for each $i = 0 \dots N - 2$.
- $1 \le Q \le 100\,000$.
- $0 \le U_j, V_j < N$ and there is an edge between nodes U_j and V_j for each $j = 0 \dots Q - 1$.
- $0 \le X_j < 2^{30}$ for each $j = 0 \dots Q - 1$.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points) Examples.

– **Subtask 2** (7 points) $N, Q \leq 100$.

– **Subtask 3** (9 points) $N \leq 2000, Q \leq 100$.

– **Subtask 4** (11 points) $N \leq 2000, Q \leq 2000$.

– **Subtask 5** (8 points) $0 \leq W_i \leq 1$, $X_j = 1$.

– **Subtask 6** (10 points) $0 \leq W_i \leq 1$, $X_j = 0$.

– **Subtask 7** (22 points) The tree is a line graph and there is an edge between nodes i and $i + 1$ for all $0 \leq i < N - 1$.

– **Subtask 8** (18 points) $0 \leq W_i \leq 1$.

– **Subtask 9** (15 points) No additional constraints.

Examples

input	output
4 0 1 1 0 2 2 0 3 3 2 0 3 4 0 2 5	9 7 15
7 0 1 0 0 2 0 1 3 1 1 4 0 2 5 0 5 6 1 5 0 2 1 1 4 1 0 2 1 2 5 1 0 1 1	2 3 5 5 9 21
8 0 1 1 0 2 1 1 3 1 1 4 1 2 5 1 5 6 0 6 7 1 4 1 3 0 5 6 0 0 2 0 6 7 0	16 11 11 5 4
6 0 1 11 1 2 6 2 3 15 3 4 13 4 5 7 3 1 2 14 2 3 10 3 4 6	93 141 114 96

input	output
10	24 14 29 17 16 28 28 36
6 4 1	
4 8 1	
4 7 1	
3 8 0	
3 9 1	
5 9 1	
0 8 1	
8 2 1	
7 1 1	
7	
7 4 0	
8 3 1	
9 3 0	
9 5 0	
4 7 1	
1 7 1	
3 9 1	
10	24048471575 27735341590
7 8 57060341	26470706958 26585373193
7 6 912175869	17345134615 17809394976
4 9 722659129	17976445112 17976445112
1 6 1070069467	
4 2 1054506724	
4 3 803713203	
0 6 1042268623	
5 0 430394330	
3 6 761326510	
7	
5 0 759019469	
6 1 737763327	
7 8 186596588	
4 3 494827354	
3 6 930475517	
4 3 389510846	
6 1 737763327	

Explanation

In the **first test case**:

- Before the first query, the cost of the tree is $cost(0,1) + cost(0,2) + cost(0,3) + cost(1,2) + cost(1,3) + cost(2,3) = 1 + 2 + 3 + (1\&2) + (1\&3) + (2\&3) = 1 + 2 + 3 + 0 + 1 + 2 = 9$.

- After the first query, the cost of the tree is $cost(0,1) + cost(0,2) + cost(0,3) + cost(1,2) + cost(1,3) + cost(2,3) = 1 + 2 + 4 + (1\&2) + (1\&4) + (2\&4) = 1 + 2 + 4 + 0 + 0 + 0 = 7$.

- After the second query, the cost of the tree is $cost(0,1) + cost(0,2) + cost(0,3) + cost(1,2) + cost(1,3) + cost(2,3) = 1 + 5 + 4 + (1\&5) + (1\&4) + (5\&4) = 1 + 5 + 4 + 1 + 0 + 4 = 15$.