Task: KOL
Dinners

XXVI OI, Stage II, Day pierwszy. Source file `kol.*` Available memory: 512 MB. *13.02.2019*

Bytie and Bytene have been a couple for many years, and over time they established an evening ritual of romantic dinners in Bitovo's magnificent yet cozy restaurants. They both work as insurance agents, who finish their work days late at random locations in town, so reaching their desired restaurant(s) at similar time always proved challenging. Lately they came to a conclusion that these escapades require too much time and money. By which they mean travel money, because a good meal is priceless to foodies like them.

Bitovo is a fairly well designed town. Its public transport network has n stops and $n-1$ segments that connect them. Each segment links two stops directly and requires a separate ticket. Every pair of stops is connected, either directly or not.

Every stop has its own restaurant nearby. The restaurants specialize in various cuisines (Mediterranean, Oriental, etc.); for convenience, we will represent the cuisines by integers from 1 to r.

Your task is to write a program that will help Bytie and Bytene choose where they should have dinner in order to minimize their overall ticket cost, given the information at which stops they finish their workday and what cuisine they fancy that particular evening.

Input

In the first line of the standard input, there are two integers n and r ($2 \leq n \leq 100\,000$, $1 \leq r \leq 100\,000$), separated by a single space, which specify, respectively, the number of stops in Bitovo and the number of different cuisines. The stops are numbered with integers from 1 to n.

The second line of input contains a sequence of n integers $t_1, t_2, \ldots, t_n$ ($1 \leq t_i \leq r$), separated by single spaces; the number t_i corresponds to the cuisine the restaurant at stop no. i specializes in.

The $n-1$ lines that follow describe the public transport network; the j-th such line contains three integers a_j, b_j, c_j ($1 \leq a_j, b_j \leq n, a_j \neq b_j, 0 \leq c_j \leq 10^6$), separated by single spaces, which indicate that the stops no. a_j and b_j are directly connected, and the ticket for that segment costs c_j.

The next line contains a single integer q ($1 \leq q \leq 100\,000$), equal to the number of days on which Bytie and Bytene will query the program for suggestions. The last q s these queries; the k-th such line has three integers p_k, q_k, s_k ($1 \leq p_k, q_k \leq n, 1 \leq s_k \leq r$), separated by single spaces, which specify the stop at which Bytie finishes his workday, the stop at which Bytene finishes hers, and the cuisine they fancy that particular evening, respectively.

Output

Exactly q lines should be printed to the standard output; the k-th of these should hold one integer: the minimum cost of reaching a restaurant with the cuisine of the couple's choosing on the k-th evening. If no such restaurant exists, `-1` should be printed instead.

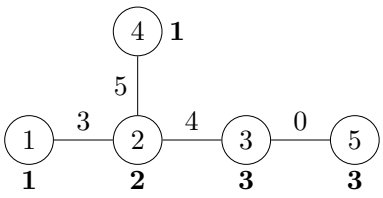
Example

For the following input data:

```
5 4
1 2 3 1 3
1 2 3
2 3 4
2 4 5
3 5 0
4
1 3 3
1 4 2
1 5 1
3 3 4
```

the correct result is:

```
7
8
7
-1
```



Explanation for the example:
The encircled numbers are stop (or restaurant) numbers, the bold numbers give the cuisine the restaurants specialize in, the edges depict the public transport segments which directly connect the stops, and the numbers next to the edges give the ticket costs.

- On the first evening, the couple wants a restaurant with the third cuisine. There are two such restaurants: at stops no. 3 and 5. For either restaurant, reaching it would cost Bytie 7, and Bytene 0. The total cost is thus 7.
- On the second evening, Bytie and Bytene choose a restaurant with the second cuisine. The only such restaurant is at stop no. 2, and reaching it costs $3 + 5 = 8$.
- On the first evening, the couple wants a restaurant with the first cuisine. Again, there are two, at stops no. 1 and 4. Reaching those costs $0 + 7 = 7$ and $8 + 9 = 17$ respectively.
- Unfortunately, there is no restaurant with the fourth cuisine yet in Bitovo, hence the `-1` answer to the last query.

Sample grading tests:

`1ocen`: 10 randomly connected stops, $r = 2$, all possible queries;

`2ocen`: 100 stops, with a restaurant with the i-th cuisine at stop no. i; two queries per cuisine;

`3ocen`: 10 000 stops along a line, all restaurant serve cuisine no. 1; 10 queries, i-th of which has Bytie and Bytene at stops i and n.

Grading

The set of tests consists of the following subsets. Within each subset, there may be several unit tests.
Time limits for each subset are published in SIO.

Subset	Condition	Score
1	$n, q, r \leq 100$	7
2	$n, q \leq 1000$	9
3	$n \leq 1000$	19
4	$r = 1$	10
5	$r = n$, exactly one restaurant per cuisine	10
6	$r = 2$	10
7	no further conditions	35