Lupin is back in business. Well, being the world's greatest thief, he is back in the business of making 'easy' money. He is grateful because last year, thanks to you and your folks in the Colombian programming contest, the bank robbery went smooth as silk. Money has ran out and this year he is planning a massive cryptocurrency heist using a tailored-designed ransomware known as *FujikoMine*. His right-hand man and closest ally Daisuke Jigen has identified the amount and distribution of cryptocurrency in the web.

The web setup is a tree with two different types of nodes: *(re-)transmission* and *bridge* nodes. A link between two nodes u and v is annotated with the (potential) number of cryptocurrency that can be stolen if both nodes u and v are infected with *FujikoMine*. There are some important observations about the ransomware:

- It can infect both transmission and bridge nodes, and attacks as many nodes as possible. However, since transmission nodes — unlike bridge nodes — are usually under heavy scrutiny by sysadmins, the plan is to limit the amount of transmission nodes to infect while maximizing the potential of the attack.

- Attacks are only effective on paths starting and ending with infected transmission nodes, where all intermediate nodes in the path are also infected.

- If u and v are infected transmission nodes and there is a path from u to v, then the number of cryptocurrency that can be stolen by an attack from u to v is the sum of cryptocurrency that can be stolen in the path from u to v, if in the path all nodes are infected with *FujikoMine* (otherwise, the sum is 0).

- The paths induced by the infected transmission nodes must form a subtree of the web setup.

- Jigen-san has assured Lupin that each node in the web setup has at most three children.

As an example, consider the depicted web setup with 9 transmission nodes (labeled from a to i) and 7 bridge nodes (whose labels have been omitted).
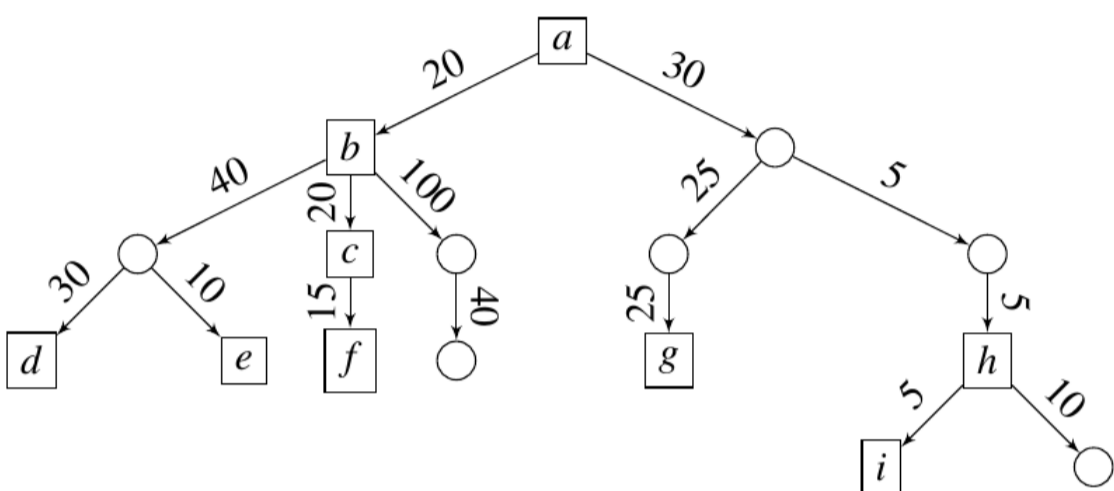
- If the plan is to infect two transmission nodes and any number of bridge nodes, the best option is to infect a and g because $30 + 25 + 25 = 80$ cryptocurrency can be stolen. Infecting a, d or g, h yields 0 cryptocurrency. In the first case, the path from a to d includes the uninfected transmission node b (otherwise, there would be three transmission nodes infected). In the second case, there is no path from g to h or from h to g.

- If the plan is to infect three nodes, the greatest potential is 100 and it can be achieved by infecting, e.g., a, b, g. Infecting nodes a, b, d, nodes b, c, d, or nodes a, g, h, only has potential 90.

Given a distribution of cryptocurrency in the web and the number of transmission nodes to infect, your task is to aid Lupin in using *FujikoMine* by computing the maximum number of cryptocurrency that can be stolen.

Input

The input consists of several test cases. In the first line of a test case there are three blank-separated integers n, m, q ($2 \leq n \leq 500$, $1 \leq m \leq n$, and $1 \leq q \leq 100$), where n is the number of nodes of the web setup, m is the number of transmission nodes, and q is the number of queries. The nodes of the web setup are identified with numbers from 0 to $n-1$. Each of the next $n-1$ lines contains three blank-separated integers u, v, w ($0 \leq u, v < n$ and $0 \leq w \leq 500$) indicating that u is a parent of v and that there are w cryptocurrency that can be stolen by attacking u and v. Next comes a line with a sequence of m blank-separated pairwise-distinct node labels t_i ($0 \leq t_i < n$ for each $1 \leq i \leq m$) identifying the transmission nodes in the web setup. Finally, comes a line with a sequence of q blank-separated integers x_j ($0 \leq x_j \leq n$ for each $1 \leq j \leq q$) indicating the number of transmission nodes to infect with *FujikoMine*.

You can assume that the given setup corresponds to a valid web setup, i.e., that the given graph is a tree with each node having at most three children.

Output

For each web setup and each query x_j, print a single line with the maximum number of cryptocurrency that can be stolen by infecting exactly x_j transmission nodes and any amount of bridge nodes.

Sample Input

```
16 9 5
0 1 20
0 9 30
1 10 40
1 2 20
1 11 100
9 12 25
9 13 5
10 3 30
10 4 10
2 5 15
11 14 40
12 6 25
13 7 5
7 8 5
7 15 10
0 1 2 3 4 5 6 7 8
1 2 3 4 16
2 1 1
1 0 2
1
1
```

Sample Output

```
0
80
100
170
0
0
```