G. Harry Vs Voldemort

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

After destroying all of Voldemort's Horcruxes, Harry and Voldemort are up for the final battle. They each cast spells from their wands and the spells collide.

The battle scene is Hogwarts, which can be represented in the form of a tree. There are, in total, n places in Hogwarts joined using $n - 1$ undirected roads.

Ron, who was viewing this battle between Harry and Voldemort, wondered how many triplets of places (u, v, w) are there such that if Harry is standing at place u and Voldemort is standing at place v, their spells collide at a place w. This is possible for a triplet only when u, v and w are distinct, and there exist paths from u to w and from v to w which do not pass through the same roads.

Now, due to the battle havoc, new paths are being added all the time. You have to tell Ron the answer after each addition.

Formally, you are given a tree with n vertices and $n - 1$ edges. q new edges are being added between the nodes of the tree. After each addition you need to tell the number of triplets (u, v, w) such that u, v and w are distinct and there exist two paths, one between u and w, another between v and w such that these paths do not have an edge in common.

Input

First line contains an integer n $(1 \le n \le 10^5)$, the number of places in Hogwarts.

Each of the next $n - 1$ lines contains two space separated integers u and v $(1 \le u, v \le n)$ indicating a road between places u and v. It is guaranteed that the given roads form a connected tree.

Next line contains a single integer q $(1 \le q \le 10^5)$, the number of new edges being added.

Each of the next q lines contains two space separated integers u and v $(1 \le u, v \le n)$ representing the new road being added.

Note that it is possible that a newly added road connects places that were connected by a road before. Also, a newly added road may connect a place to itself.

Output

In the first line print the value for the number of triplets before any changes occurred.

After that print q lines, a single integer ans_i in each line containing the value for the number of triplets after i-th edge addition.

Examples

input

```
3
1 2
2 3
1
2 3
```

output

```
2
4
```

input

```
4
1 2
2 3
2 4
2
1 4
3 4
```

output

```
6
18
24
```

input

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

output

```
20
60
```

Note

In the first sample case, for the initial tree, we have $(1, 3, 2)$ and $(3, 1, 2)$ as the only possible triplets (u, v, w).

After addition of edge from 2 to 3, we have $(1, 3, 2)$, $(3, 1, 2)$, $(1, 2, 3)$ and $(2, 1, 3)$ as the possible triplets.