I: Balanced Tree Path

Time Limit: 5 seconds, Memory limit: 2G

You are given a tree where each node is annotated with a character from `()[]{}`. A path is a sequence of one or more nodes where no node is repeated and every pair of adjacent nodes is connected with an edge. A path is balanced if the characters at each node, when concatenated, form a balanced string. A string is balanced if it satisfies the following definition:

- An empty string is balanced.
- If s is a balanced string, then `(`s`)`, `[`s`]`, and `{`s`}` are balanced strings.
- if a and b are balanced strings, then ab (a concatenated with b) is a balanced string.

Compute the number of balanced paths over the entire tree.

Input

The first line of input contains a single integer n ($2 \leq n \leq 5 \cdot 10^3$).

The next line contains a string of n characters, where each character is one of `()[]{}`.

Each of the next $n-1$ lines contains two integers, u and v ($1 \leq u < v \leq n$), indicating that nodes u and v are connected with an edge. It is guaranteed the graph is a tree.

Output

Output a single integer, which is the number of balanced paths over the entire tree.

Sample Input 1	Sample Output 1
4 ()() 1 2 2 3 3 4	4

Sample Input 2

```
4
[[]]
1 2
2 3
3 4
```

Sample Output 2

```
2
```

Sample Input 3

```
6
([]{})
1 2
2 3
3 4
4 5
5 6
```

Sample Output 3

```
4
```