Gems

The Gem-Toys Company asked you to solve the following problem. You are given a connected acyclic graph, i.e. a set of vertices connected by edges in such a way that from each vertex you can reach all the others vertices by traversing the edges, and it does not contain a loop. The Gem-Toys Company is going to produce jewelry models of such graphs. Vertices will be made of gems and edges will be made of gold string. It is required that adjacent vertices are made of different kinds of gems. For each positive integer p there is exactly one kind of gems with price p. The main cost is a cost of gems. Your task is to write a program computing the minimal total price of gems needed to make the model.

Input

Your program should read data from the standard input. The first line contains one positive integer n, number of vertices $1 \leq n \leq 10\,000$. The vertices are numbered from 1 to n. The following $n-1$ lines describe the edges, one per line. Each of these lines contains a pair of integers a and b separated by a space, $1 \leq a, b \leq n$, $a \neq b$. Such a pair represents an edge connecting vertices a and b.

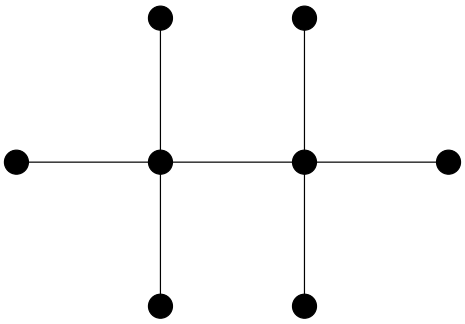
Output

Your program should write just one integer to the standard output: the minimal cost of gems needed to make the model.

Example

For the input:

```
8
1 2
3 1
1 4
5 6
1 5
5 7
5 8
```



the correct result is:

```
11
```

Solution

This task can be solved by dynamic programming. Let us root the tree by choosing one of the leaves as a root. Then we can process the tree in a bottom-up order. For each vertex we should compute two solutions for a sub-tree rooted in the given vertex:

- the optimal solution, together with the gem g used for the given vertex,
- the optimal solution, assuming, that the given vertex is modeled by a gem different from g.

If this task is too easy, it can be made more difficult by allowing an exception: there can be at most one pair of adjacent vertices modeled by the same kind of gems. The solution schema remains unchanged, but it gets more complicated.