

White-Black-Tree

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 1024 mebibytes

There is a tree with N vertices. Vertices are numbered from 1 to N .

Each vertex is colored either white or black. You can swap the colors of two adjacent vertices any number of times you want. A **final tree** is any tree formed by completing a finite number of swaps in the original tree. The **swap cost** of a final tree is equal to the number of swaps you have made.

A white tree is the connected subgraph of the final tree which connects all the white vertices of the final tree using the minimum number of edges. Likewise, a black tree is the connected subgraph of the final tree which connects all the black vertices of the final tree using the minimum number of edges.

The **edge cost** of a final tree is equal to the sum of the number of edges in the white tree and the number of edges in the black tree. Note that the edge cost is calculated only after all swaps have been completed.

The total cost of a final tree is the sum of its swap cost and edge cost.

Find the minimum possible total cost of a final tree.

Input

The first line of input contains the number of vertices N .

The second line of input contains a string s of length N consisting only of W and B. If the i -th character of s is W, vertex i is white, and if the i -th character is B, vertex i is black. It is guaranteed that there are at least one W and at least one B in the string s .

Each of the following $N - 1$ lines contains u and v denoting that there is an edge connecting vertex u and vertex v .

It is guaranteed that the input forms a valid tree.

Output

The first line of output should contain the minimum possible total cost of a final tree.

Scoring

- $2 \leq N \leq 5 \times 10^5$
- $1 \leq u < v \leq N$

Examples

standard input	standard output
4 WBBB 1 2 2 3 2 4	3
8 WBWWBBB 1 2 2 3 3 4 3 5 1 6 6 7 6 8	7