# Christmas Balls (`christmasballs`)

Christmas is coming, and Alessandro wants to buy a Christmas tree to decorate his living room. When he heard that in Pordenone a new tree shop just opened, he immediately decided to go there to poke around. This shop has a huge tree decorated with $N$ balls, one at each branching point (the root, node 0, lies at the base of the tree). The balls are quite spectacular: Alessandro noticed that they come in $C$ different colors in total.

The peculiarity of this shop is that you can buy the whole tree, or cut a branch and buy just a *subtree*! You can make the cut just below a ball, so that you take that ball and the subtree rooted (growing from) there.

But Alessandro doesn't want just a random tree, he wants the *nicest* one! His taste is in fact quite peculiar: he appreciates when the number of balls of each color is well distributed. More precisely, if we denote with $x$ the number of balls painted with (one of) the most frequent color in a subtree, he wants to maximize the number of colors that appear in $x$ balls each.

Help Alessandro find where to cut the tree by printing the maximum amount of most-frequent colors he can get.

> ☞ Among the attachments of this task you may find a template file `christmasballs.*` with a sample incomplete implementation.

## Input

The first line contains the only integer $N$, the number of nodes of the tree. The second line contains $N$ integers $C_i$, the colors of each ball. The third line contains $N-1$ integers: $P_i$ (with $1 \leq i < N$) is the index of the parent of the $i$-th node. The parent of the root node ($P_0$) is not defined and therefore **not present in the input**.

## Output

You need to write a single line with an integer: the number of colors with the highest frequency in the optimal subtree.

## Constraints

- $1 \leq N \leq 100\,000$.
- $0 \leq C_i < N$, for all $i = 0 \ldots N-1$.
- $0 \leq P_i < i$, for all $i = 1 \ldots N-1$.

## Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points) Examples.

– **Subtask 2** (7 points) $N \leq 1000$ and the tree is just a line: $P_i = i - 1$ for all $i = 1 \ldots N-1$.

– **Subtask 3** (9 points) The tree is just a line: $P_i = i - 1$ for all $i = 1 \ldots N-1$.

– **Subtask 4** (15 points) $N \leq 1000$ and $C \leq 2$.

– **Subtask 5** (19 points) $N \leq 1000$.

– **Subtask 6** (17 points) $C \leq 2$.

– **Subtask 7** (33 points) No additional limitations.

## Examples

| input | output |
|---|---|
| 8<br>1 2 0 2 0 0 1 1<br>0 0 1 1 3 4 4 | 3 |
| 5<br>0 1 1 0 0<br>0 1 2 3 | 2 |

## Explanation

In the figures below, the small number at the top left is the number representing the color. We arbitrarily associate color 0 with green, color 1 with blue, and color 2 with orange.

In the **first sample case** there are 8 nodes in 3 different colors. The solution is to cut the tree and take the subtree rooted at node 1. This way we are left with 6 nodes: the most frequent color has 2 balls, and there are 3 colors with that many balls.

If, for example, we keep the entire tree, the most frequent color appears with 3 balls, but there are only 2 colors with 3 balls, and it would be worse than the aforementioned solution.

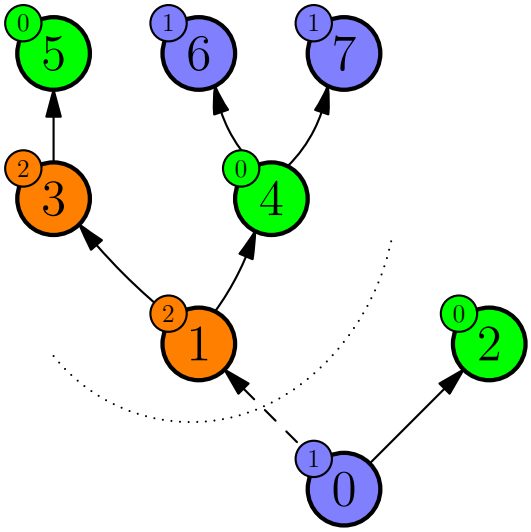


Figure 1: First sample case.

In the **second sample case** the tree forms a line with 5 nodes. The optimal solution is to cut the subtree rooted at node 1. Then, the most frequent color appears with 2 balls, and there are 2 colors with that many balls.

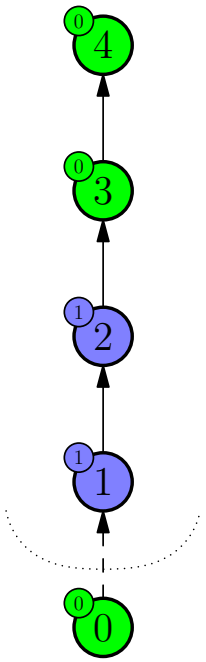


Figure 2: Second sample case.