3004. Maximum Subtree of the Same Color Premium

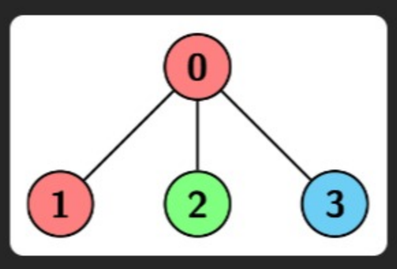
Medium | Topics | Companies | Hint

You are given a 2D integer array `edges` representing a tree with `n` nodes, numbered from `0` to `n - 1`, rooted at node `0`, where `edges[i] = [u_i, v_i]` means there is an edge between the nodes `v_i` and `u_i`.

You are also given a **0-indexed** integer array `colors` of size `n`, where `colors[i]` is the color assigned to node `i`.

We want to find a node `v` such that every node in the subtree of `v` has the **same** color.

Return *the size of such subtree with the **maximum** number of nodes possible.*

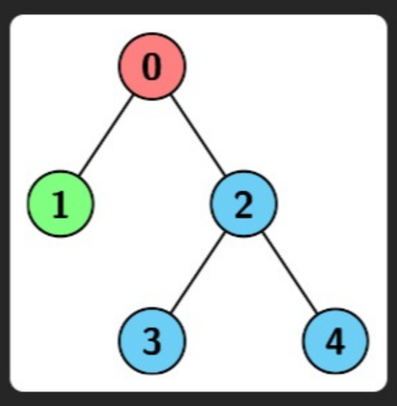


Example 1:

```
Input: edges = [[0,1],[0,2],[0,3]], colors = [1,1,2,3]
Output: 1
Explanation: Each color is represented as: 1 -> Red, 2 -> Green, 3 -> Blue. We can see that the subtree rooted at node 0 has children with different colors. Any other subtree is of the same color and has a size of 1. Hence, we return 1.
```

Example 2:

```
Input: edges = [[0,1],[0,2],[0,3]], colors = [1,1,1,1]
Output: 4
Explanation: The whole tree has the same color, and the subtree rooted at node 0 has the most number of nodes which is 4. Hence, we return 4.
```



Example 3:

```
Input: edges = [[0,1],[0,2],[2,3],[2,4]], colors = [1,2,3,3,3]
Output: 3
Explanation: Each color is represented as: 1 -> Red, 2 -> Green, 3 -> Blue. We can see that the subtree rooted at node 0 has children with different colors. Any other subtree is of the same color, but the subtree rooted at node 2 has a size of 3 which is the maximum. Hence, we return 3.
```

Constraints:

- `n == edges.length + 1`
- `1 <= n <= 5 * 10^4`
- `edges[i] == [u_i, v_i]`
- `0 <= u_i, v_i < n`
- `colors.length == n`
- `1 <= colors[i] <= 10^5`
- The input is generated such that the graph represented by `edges` is a tree.

Seen this question in a real interview before? 1/5

Yes No

Accepted 1.5K | Submissions 2.4K | Acceptance Rate 63.5%

Topics

Array Dynamic Programming Tree Depth-First Search

Companies

0 - 3 months

BlackRock 10

0 - 6 months

Microsoft 2

Hint 1

For each node, define a `flag[v]` indicating that the subtree of this node contains only one color or not.

Hint 2

In the DFS process, when you call `dfs(u)` from node `v`, after that DFS of `u` has finished, check if `flag[u] == false`, then `flag[v]` is also `false`.

Hint 3

Also if `color[v] != color[u]`, `flag[v]` becomes `false`.

Discussion (3)