

Problem 3425: Longest Special Path

Problem Information

Difficulty: Hard

Acceptance Rate: 21.40%

Paid Only: No

Tags: Array, Hash Table, Tree, Depth-First Search, Prefix Sum

Problem Description

You are given an undirected tree rooted at node `0` with `n` nodes numbered from `0` to `n - 1`, represented by a 2D array `edges` of length `n - 1`, where `edges[i] = [ui, vi, lengthi]` indicates an edge between nodes `ui` and `vi` with length `lengthi`. You are also given an integer array `nums`, where `nums[i]` represents the value at node `i`.

A **special path** is defined as a **downward** path from an ancestor node to a descendant node such that all the values of the nodes in that path are **unique**.

Note that a path may start and end at the same node.

Return an array `result` of size 2, where `result[0]` is the **length** of the **longest** special path, and `result[1]` is the **minimum** number of nodes in all **possible** **longest** special paths.

Example 1:

Input: edges = [[0,1,2],[1,2,3],[1,3,5],[1,4,4],[2,5,6]], nums = [2,1,2,1,3,1]

Output: [6,2]

Explanation:

In the image below, nodes are colored by their corresponding values in `nums`

The longest special paths are `2 -> 5` and `0 -> 1 -> 4`, both having a length of 6. The minimum number of nodes across all longest special paths is 2.

Example 2:

Input: edges = [[1,0,8]], nums = [2,2]

Output: [0,1]

Explanation:

The longest special paths are `0` and `1`, both having a length of 0. The minimum number of nodes across all longest special paths is 1.

Constraints:

$2 \leq n \leq 5 \cdot 10^4$ * `edges.length == n - 1` * `edges[i].length == 3` * $0 \leq u_i, v_i < n$ * $1 \leq \text{length}_i \leq 103$ * `nums.length == n` * $0 \leq \text{nums}[i] \leq 5 \cdot 10^4$ * The input is generated such that `edges` represents a valid tree.

Code Snippets

C++:

```
class Solution {
public:
    vector<int> longestSpecialPath(vector<vector<int>>& edges, vector<int>& nums)
    {

    }

};
```

Java:

```
class Solution {
    public int[] longestSpecialPath(int[][] edges, int[] nums) {

    }
}
```

```
}
```

Python3:

```
class Solution:
    def longestSpecialPath(self, edges: List[List[int]], nums: List[int]) ->
        List[int]:
```