2925. Maximum Score After Applying Operations on a Tree

My Submissions (/contest/weekly-contest-370/problems/maximum-score-after-applying-operations-on-a-tree/submissions/)

Back to Contest (/contest/weekly-contest-370/)

User Accepted:	2391
User Tried:	3379
Total Accepted:	2512
Total Submissions:	6653
Difficulty:	Medium

There is an undirected tree with `n` nodes labeled from `0` to `n - 1`, and rooted at node `0`. You are given a 2D integer array `edges` of length `n - 1`, where `edges[i] = [`a_i`, `b_i`]` indicates that there is an edge between nodes a_i and b_i in the tree.

You are also given a **0-indexed** integer array `values` of length `n`, where `values[i]` is the **value** associated with the i^{th} node.

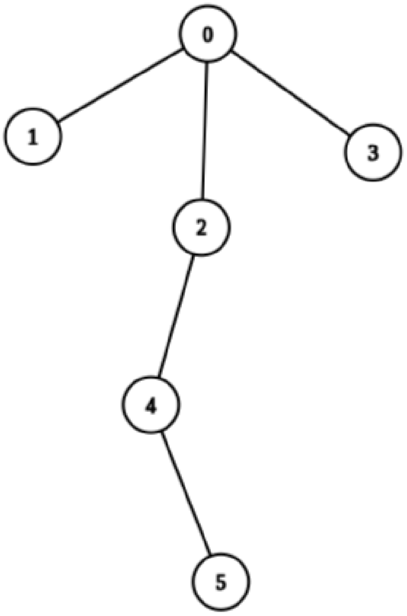
You start with a score of `0`. In one operation, you can:

- Pick any node `i`.
- Add `values[i]` to your score.
- Set `values[i]` to `0`.

A tree is **healthy** if the sum of values on the path from the root to any leaf node is different than zero.

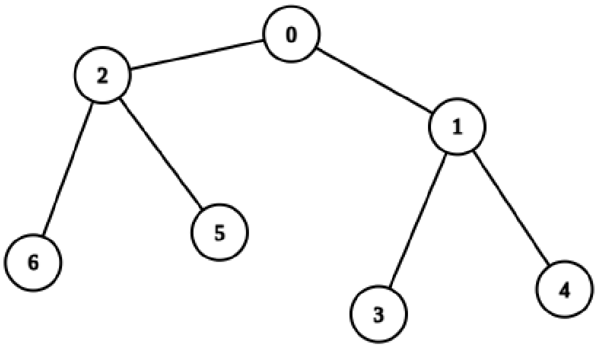
Return *the **maximum score** you can obtain after performing these operations on the tree any number of times so that it remains **healthy**.*

Example 1:



```
Input: edges = [[0,1],[0,2],[0,3],[2,4],[4,5]], values = [5,2,5,2,1,1]
Output: 11
Explanation: We can choose nodes 1, 2, 3, 4, and 5. The value of the root is non-zero. Hence, the sum of values on t
It can be shown that 11 is the maximum score obtainable after any number of operations on the tree.
```

Example 2:



```
Input: edges = [[0,1],[0,2],[1,3],[1,4],[2,5],[2,6]], values = [20,10,9,7,4,3,5]
Output: 40
Explanation: We can choose nodes 0, 2, 3, and 4.
- The sum of values on the path from 0 to 4 is equal to 10.
- The sum of values on the path from 0 to 3 is equal to 10.
- The sum of values on the path from 0 to 5 is equal to 3.
- The sum of values on the path from 0 to 6 is equal to 5.
Therefore, the tree is healthy and the score is values[0] + values[2] + values[3] + values[4] = 40.
It can be shown that 40 is the maximum score obtainable after any number of operations on the tree.
```

Constraints:

- 2 <= n <= 2 * 10^4
- edges.length == n - 1
- edges[i].length == 2
- 0 <= a_i, b_i < n
- values.length == n
- 1 <= values[i] <= 10^9
- The input is generated such that edges represents a valid tree.

Discuss (https://leetcode.com/problems/maximum-score-after-applying-operations-on-a-tree/discuss)

JavaScript

```javascript
const initializeGraph = (n) => { let g = []; for (let i = 0; i < n; i++) { g.push([]); } return g; };
const packUG = (g, edges) => { for (const [u, v] of edges) { g[u].push(v); g[v].push(u); } };
const sm = (a) => a.reduce(((x, y) => x + y), 0);

let g, vals, res;
const maximumScoreAfterOperations = (edges, values) => {
    let n = values.length;
    g = initializeGraph(n), vals = values, res = sm(vals);
    packUG(g, edges);
    res -= tree_dp(0, -1);;
    return res;
};

const tree_dp = (cur, par) => { // 计算以cur为根的子树是健康时，失去的最小分数
    if (cur != 0 && g[cur].length == 1) return vals[cur]; // cur is leaf, cur != 0 避免误把根节点当作叶子
    let subTreeSum = 0; // loss 不选vals[x]
    for (const child of g[cur]) {
        if (child != par) {
            subTreeSum += tree_dp(child, cur); // 计算以child为根的子树是健康时，失去的最小分数
        }
    }
    return Math.min(vals[cur], subTreeSum); // 选/不选vals[x]，取最小值
};

```

□ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/1097184709/) More Details (/submissions/detail/1097184709/)

Share your acceptance!

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)