(/) Explore(/explore/) Problems(/problemset/all/) Interview New (/contest/) Contest Discuss(/discuss/) Store ☆ Premium (/subscribe?ref=nb_npl) 0 (/problems/longest-cycle-in-a-graph/)

6356. Collect Coins in a Tree

My Submissions (/contest/weekly-contest-338/problems/collect-coins-in-a-tree/submissions/) Back to Contest (/contest/weekly-contest-338/)

User Accepted:	4
User Tried:	15
Total Accepted:	4
Total Submissions:	23
Difficulty:	Hard

There exists an undirected and unrooted tree with `n` nodes indexed from `0` to `n - 1`. You are given an integer `n` and 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 an array `coins` of size `n` where `coins[i]` can be either `0` or `1`, where `1` indicates the presence of a coin in the vertex `i`.

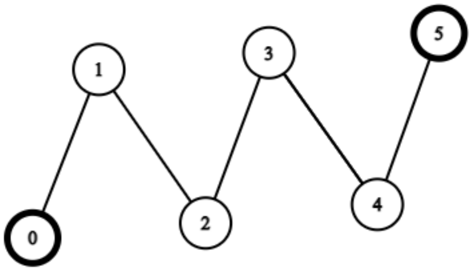
Initially, you choose to start at any vertex in the tree. Then, you can perform the following operations any number of times:

- Collect all the coins that are at a distance of at most `2` from the current vertex, or
- Move to any adjacent vertex in the tree.

Find *the minimum number of edges you need to go through to collect all the coins and go back to the initial vertex*.

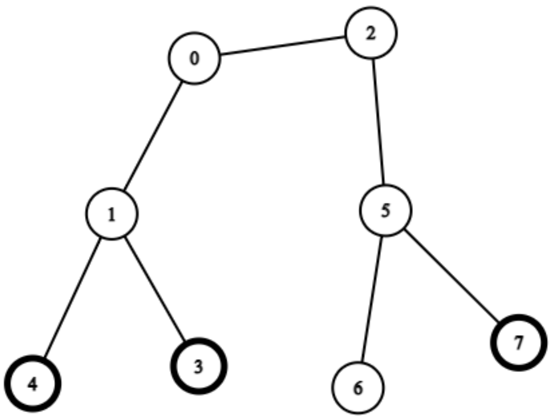
Note that if you pass an edge several times, you need to count it into the answer several times.

Example 1:



```
Input: coins = [1,0,0,0,0,1], edges = [[0,1],[1,2],[2,3],[3,4],[4,5]]
Output: 2
Explanation: Start at vertex 2, collect the coin at vertex 0, move to vertex 3, collect the coin at vertex 5 then move back to
```

Example 2:



```
Input: coins = [0,0,0,1,1,0,0,1], edges = [[0,1],[0,2],[1,3],[1,4],[2,5],[5,6],[5,7]]
Output: 2
Explanation: Start at vertex 0, collect the coins at vertices 4 and 3, move to vertex 2,  collect the coin at vertex 7, then mo
```

Constraints:

- `n == coins.length`
- `1 <= n <= 3 * ` 10^4
- `0 <= coins[i] <= 1`
- `edges.length == n - 1`
- `edges[i].length == 2`
- `0 <=` a_i, b_i `< n`
- a_i `!=` b_i
- `edges` represents a valid tree.

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 a, cum, res, v, sum, g;
const collectTheCoins = (coins, edges) => {
    let n = coins.length;
    g = initializeGraph(n), a = coins, res = 0, cum = Array(n), v = Array(n), sum = sm(a);
    packUG(g, edges);
    for (let i = 0; i < n; i++) {
        cum[i] = a[i];
        for (const child of g[i]) cum[i] += a[child];
    }
    tree_dp(0, 0);
    return res;
};

const tree_dp = (cur, par) => {
    v[cur] = a[cur];
    for (const child of g[cur]) {
        if (child != par) {
            v[cur] += tree_dp(child, cur);
        }
    }
    if (cur != par) {
        let x = v[cur] + cum[par] - a[cur];
        let y = (sum - v[cur]) + cum[cur] - a[par];
        if (x != sum && y != sum) res += 2;
    }
    return v[cur];
};
```

☐ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/923565657/) More Details (/submissions/detail/923565657/)

Share your acceptance!

2

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

United States (/region)