# 6294. Difference Between Maximum and Minimum Price Sum

My Submissions (/contest/weekly-contest-328/problems/difference-between-maximum-and-minimum-price-sum/submissions/)

Back to Contest (/contest/weekly-contest-328/)

There exists an undirected and initially unrooted tree with `n` nodes indexed from `0` to `n - 1`. You are given the 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.

Each node has an associated price. You are given an integer array `price`, where `price[i]` is the price of the `i^th` node.

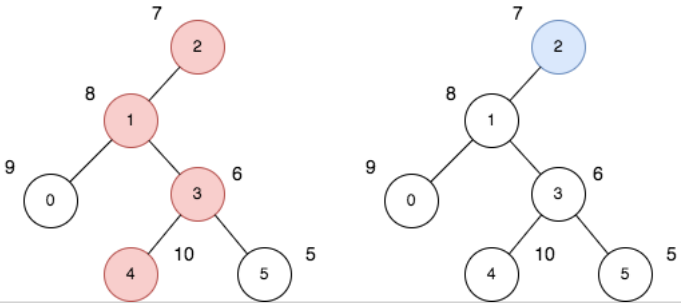
The **price sum** of a given path is the sum of the prices of all nodes lying on that path.

The tree can be rooted at any node `root` of your choice. The incurred **cost** after choosing `root` is the difference between the maximum and minimum **price sum** amongst all paths starting at `root`.

Return *the **maximum** possible **cost** amongst all possible root choices*.

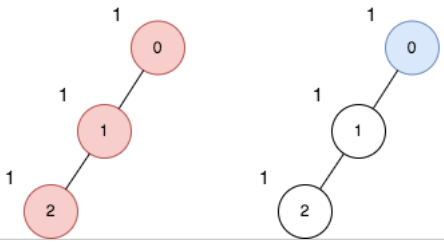
| | |
|---|---|
| User Accepted: | 0 |
| User Tried: | 0 |
| Total Accepted: | 0 |
| Total Submissions: | 0 |
| Difficulty: | Hard |

**Example 1:**




```
Input: n = 6, edges = [[0,1],[1,2],[1,3],[3,4],[3,5]], price = [9,8,7,6,10,5]
Output: 24
Explanation: The diagram above denotes the tree after rooting it at node 2. The first part (colored in red) shows the path with
- The first path contains nodes [2,1,3,4]: the prices are [7,8,6,10], and the sum of the prices is 31.
- The second path contains the node [2] with the price [7].
The difference between the maximum and minimum price sum is 24. It can be proved that 24 is the maximum cost.
```

**Example 2:**




```
Input: n = 3, edges = [[0,1],[1,2]], price = [1,1,1]
Output: 2
Explanation: The diagram above denotes the tree after rooting it at node 0. The first part (colored in red) shows the path with
- The first path contains nodes [0,1,2]: the prices are [1,1,1], and the sum of the prices is 3.
- The second path contains node [0] with a price [1].
The difference between the maximum and minimum price sum is 2. It can be proved that 2 is the maximum cost.
```

**Constraints:**

- `1 <= n <= 10^5`
- `edges.length == n - 1`
- `0 <= a_i, b_i <= n - 1`
- `edges` represents a valid tree.

- price.length == n
- 1 <= price[i] <= $10^5$

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); } };

let g, p, memo;
const maxOutput = (n, edges, price) => {
    g = initializeGraph(n), p = price, memo = new Map();
    packUG(g, edges);
    let res = [];
    for (let i = 0; i < n; i++) {
        let max = dfs(i, -1), v = max - p[i];
        res.push(v);
    }
    return Math.max(...res);
}

const dfs = (cur, par) => {
    let ke = cur + ' ' + par;
    if (memo.has(ke)) return memo.get(ke);
    let d = [0];
    for (const child of g[cur]) {
        if (child != par) {
            let v = dfs(child, cur);
            d.push(v);
        }
    }
    let res = p[cur] + Math.max(...d);
    memo.set(ke, res);
    return res;
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/878440645/) | More Details > (/submissions/detail/878440645/)

Share your acceptance!