5406. Minimum Time to Collect All Apples in a Tree

/ Submissions (/contest/weekly-contest-188/problems/minimum-time-to-collect-all-apples-in-a-tree/submissions/)

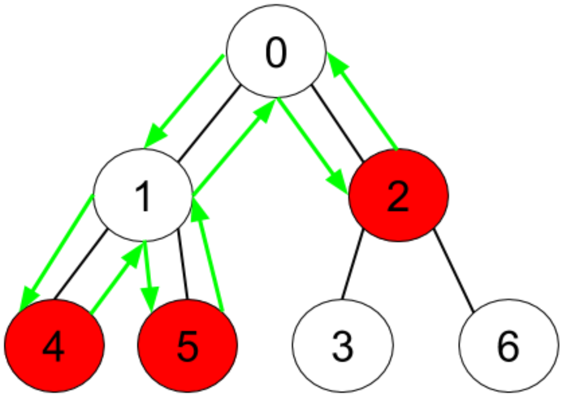
ck to Contest (/contest/weekly-contest-188/)

Given an undirected tree consisting of `n` vertices numbered from 0 to `n-1`, which has some apples in their vertices. You spend 1 second to walk over one edge of the tree. *Return the minimum time in seconds you have to spend in order to collect all apples in the tree starting at **vertex 0** and coming back to this vertex.*

The edges of the undirected tree are given in the array `edges`, where `edges[i] = [`$\texttt{from}_i$`, `$\texttt{to}_i$`]` means that exists an edge connecting the vertices $\texttt{from}_i$ and $\texttt{to}_i$. Additionally, there is a boolean array `hasApple`, where `hasApple[i] = true` means that vertex `i` has an apple, otherwise, it does not have any apple.

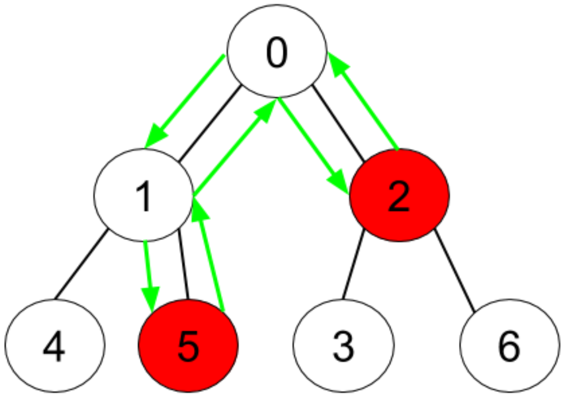
User Accepted:	2305
User Tried:	2771
Total Accepted:	2350
Total Submissions:	4053
Difficulty:	Medium

Example 1:



```
Input: n = 7, edges = [[0,1],[0,2],[1,4],[1,5],[2,3],[2,6]], hasApple = [false,false,true,
Output: 8
Explanation: The figure above represents the given tree where red vertices have an apple.
```

Example 2:



```
Input: n = 7, edges = [[0,1],[0,2],[1,4],[1,5],[2,3],[2,6]], hasApple = [false,false,true,
Output: 6
Explanation: The figure above represents the given tree where red vertices have an apple.
```

Example 3:

```
Input: n = 7, edges = [[0,1],[0,2],[1,4],[1,5],[2,3],[2,6]], hasApple = [false,false,false
Output: 0
```

Constraints:

- `1 <= n <= 10^5`
- `edges.length == n-1`
- `edges[i].length == 2`
- $0 <= from_i, to_i <= n-1$
- $from_i < to_i$
- `hasApple.length == n`

JavaScript

```
/**
 * @param {number} n
 * @param {number[][]} edges
 * @param {boolean[]} hasApple
 * @return {number}
 */
var minTime = function(n, edges, hasApple) {

};
```