Problem 1130: Minimum Cost Tree From Leaf Values

Problem Information

Difficulty: Medium
Acceptance Rate: 67.73%
Paid Only: No
Tags: Array, Dynamic Programming, Stack, Greedy, Monotonic Stack

Problem Description

Given an array `arr` of positive integers, consider all binary trees such that:

* Each node has either `0` or `2` children; * The values of `arr` correspond to the values of each **leaf** in an in-order traversal of the tree. * The value of each non-leaf node is equal to the product of the largest leaf value in its left and right subtree, respectively.

Among all possible binary trees considered, return _the smallest possible sum of the values of each non-leaf node_. It is guaranteed this sum fits into a **32-bit** integer.

A node is a **leaf** if and only if it has zero children.

Example 1:

![](https://assets.leetcode.com/uploads/2021/08/10/tree1.jpg)

Input: arr = [6,2,4] **Output:** 32 **Explanation:** There are two possible trees shown. The first has a non-leaf node sum 36, and the second has non-leaf node sum 32.

Example 2:

![](https://assets.leetcode.com/uploads/2021/08/10/tree2.jpg)

Input: arr = [4,11] **Output:** 44

Constraints:

* `2 <= arr.length <= 40` * `1 <= arr[i] <= 15` * It is guaranteed that the answer fits into a **32-bit** signed integer (i.e., it is less than 231).

Code Snippets

C++:

```
class Solution {
public:
int mctFromLeafValues(vector<int>& arr) {

}
};
```

Java:

```
class Solution {
public int mctFromLeafValues(int[] arr) {

}
}
```

Python3:

```
class Solution:
def mctFromLeafValues(self, arr: List[int]) -> int:
```