

Problem 823: Binary Trees With Factors

Problem Information

Difficulty: Medium

Acceptance Rate: 53.03%

Paid Only: No

Tags: Array, Hash Table, Dynamic Programming, Sorting

Problem Description

Given an array of unique integers, `arr`, where each integer `arr[i]` is strictly greater than `1`.

We make a binary tree using these integers, and each number may be used for any number of times. Each non-leaf node's value should be equal to the product of the values of its children.

Return the number of binary trees we can make. The answer may be too large so return the answer **modulo** `109 + 7`.

Example 1:

Input: `arr = [2,4]` **Output:** `3` **Explanation:** We can make these trees: `[2]`, `[4]`, `[4, 2, 2]`

Example 2:

Input: `arr = [2,4,5,10]` **Output:** `7` **Explanation:** We can make these trees: `[2]`, `[4]`, `[5]`, `[10]`, `[4, 2, 2]`, `[10, 2, 5]`, `[10, 5, 2]`.

Constraints:

`1 <= arr.length <= 1000` `2 <= arr[i] <= 109` All the values of `arr` are **unique**.

Code Snippets

C++:

```
class Solution {  
public:  
    int numFactoredBinaryTrees(vector<int>& arr) {  
  
    }  
};
```

Java:

```
class Solution {  
    public int numFactoredBinaryTrees(int[] arr) {  
  
    }  
}
```

Python3:

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