

Problem 2007: Find Original Array From Doubled Array

Problem Information

Difficulty: Medium

Acceptance Rate: 40.66%

Paid Only: No

Tags: Array, Hash Table, Greedy, Sorting

Problem Description

An integer array `original` is transformed into a **doubled** array `changed` by appending **twice the value** of every element in `original`, and then randomly **shuffling** the resulting array.

Given an array `changed`, return `original` if `changed` is a **doubled** array. If `changed` is not a **doubled** array, return an empty array. The elements in `original` may be returned in **any** order.

Example 1:

Input: `changed = [1,3,4,2,6,8]` **Output:** `[1,3,4]` **Explanation:** One possible original array could be `[1,3,4]`: - Twice the value of 1 is $1 * 2 = 2$. - Twice the value of 3 is $3 * 2 = 6$. - Twice the value of 4 is $4 * 2 = 8$. Other original arrays could be `[4,3,1]` or `[3,1,4]`.

Example 2:

Input: `changed = [6,3,0,1]` **Output:** `[]` **Explanation:** `changed` is not a doubled array.

Example 3:

Input: `changed = [1]` **Output:** `[]` **Explanation:** `changed` is not a doubled array.

Constraints:

```
*`1 <= changed.length <= 105` *`0 <= changed[i] <= 105`
```

Code Snippets

C++:

```
class Solution {  
public:  
    vector<int> findOriginalArray(vector<int>& changed) {  
  
    }  
};
```

Java:

```
class Solution {  
    public int[] findOriginalArray(int[] changed) {  
  
    }  
}
```

Python3:

```
class Solution:  
    def findOriginalArray(self, changed: List[int]) -> List[int]:
```