Problem 3371: Identify the Largest Outlier in an Array

Problem Information

Difficulty: Medium
Acceptance Rate: 36.07%
Paid Only: No
Tags: Array, Hash Table, Counting, Enumeration

Problem Description

You are given an integer array `nums`. This array contains `n` elements, where **exactly** `n - 2` elements are **special****numbers**. One of the remaining **two** elements is the _sum_ of these **special numbers** , and the other is an **outlier**.

An **outlier** is defined as a number that is _neither_ one of the original special numbers _nor_ the element representing the sum of those numbers.

Note that special numbers, the sum element, and the outlier must have **distinct** indices, but _may_ share the **same** value.

Return the **largest****** potential**outlier** in `nums`.

Example 1:

Input: nums = [2,3,5,10]

Output: 10

Explanation:

The special numbers could be 2 and 3, thus making their sum 5 and the outlier 10.

Example 2:

Input: nums = [-2,-1,-3,-6,4]

Output: 4

Explanation:

The special numbers could be -2, -1, and -3, thus making their sum -6 and the outlier 4.

Example 3:

Input: nums = [1,1,1,1,1,5,5]

Output: 5

Explanation:

The special numbers could be 1, 1, 1, 1, and 1, thus making their sum 5 and the other 5 as the outlier.

Constraints:

* `3 <= nums.length <= 105` * `-1000 <= nums[i] <= 1000` * The input is generated such that at least **one** potential outlier exists in `nums`.

Code Snippets

C++:

```
class Solution {
public:
int getLargestOutlier(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public int getLargestOutlier(int[] nums) {
```

```
}
}
```

Python3:

```
class Solution:
def getLargestOutlier(self, nums: List[int]) -> int:
```