Problem 1968: Array With Elements Not Equal to Average of Neighbors

Problem Information

Difficulty: Medium
Acceptance Rate: 50.39%
Paid Only: No
Tags: Array, Greedy, Sorting

Problem Description

You are given a **0-indexed** array `nums` of **distinct** integers. You want to rearrange the elements in the array such that every element in the rearranged array is **not** equal to the **average** of its neighbors.

More formally, the rearranged array should have the property such that for every `i` in the range `1 <= i < nums.length - 1`, `(nums[i-1] + nums[i+1]) / 2` is **not** equal to `nums[i]`.

Return _**any** rearrangement of _`nums` _that meets the requirements_.

Example 1:

Input: nums = [1,2,3,4,5] **Output:** [1,2,4,5,3] **Explanation:** When i=1, nums[i] = 2, and the average of its neighbors is (1+4) / 2 = 2.5. When i=2, nums[i] = 4, and the average of its neighbors is (2+5) / 2 = 3.5. When i=3, nums[i] = 5, and the average of its neighbors is (4+3) / 2 = 3.5.

Example 2:

Input: nums = [6,2,0,9,7] **Output:** [9,7,6,2,0] **Explanation:** When i=1, nums[i] = 7, and the average of its neighbors is (9+6) / 2 = 7.5. When i=2, nums[i] = 6, and the average of its neighbors is (7+2) / 2 = 4.5. When i=3, nums[i] = 2, and the average of its neighbors is (6+0) / 2 = 3. Note that the original array [6,2,0,9,7] also satisfies the conditions.

Constraints:

* `3 <= nums.length <= 105` * `0 <= nums[i] <= 105`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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