Problem 870: Advantage Shuffle

Problem Information

Difficulty: Medium
Acceptance Rate: 53.99%
Paid Only: No
Tags: Array, Two Pointers, Greedy, Sorting

Problem Description

You are given two integer arrays `nums1` and `nums2` both of the same length. The **advantage** of `nums1` with respect to `nums2` is the number of indices `i` for which `nums1[i] > nums2[i]`.

Return _any permutation of_`nums1` _that maximizes its**advantage** with respect to _`nums2`.

Example 1:

Input: nums1 = [2,7,11,15], nums2 = [1,10,4,11] **Output:** [2,11,7,15]

Example 2:

Input: nums1 = [12,24,8,32], nums2 = [13,25,32,11] **Output:** [24,32,8,12]

Constraints:

* `1 <= nums1.length <= 105` * `nums2.length == nums1.length` * `0 <= nums1[i], nums2[i] <= 109`

Code Snippets

C++:

```
class Solution {
public:
vector<int> advantageCount(vector<int>& nums1, vector<int>& nums2) {

}
};
```

Java:

```
class Solution {
public int[] advantageCount(int[] nums1, int[] nums2) {

}
}
```

Python3:

```
class Solution:
def advantageCount(self, nums1: List[int], nums2: List[int]) -> List[int]:
```