Problem 561: Array Partition

Problem Information

Difficulty: Easy
Acceptance Rate: 81.12%
Paid Only: No
Tags: Array, Greedy, Sorting, Counting Sort

Problem Description

Given an integer array `nums` of `2n` integers, group these integers into `n` pairs `(a1, b1), (a2, b2), ..., (an, bn)` such that the sum of `min(ai, bi)` for all `i` is **maximized**. Return _the maximized sum_.

Example 1:

Input: nums = [1,4,3,2] **Output:** 4 **Explanation:** All possible pairings (ignoring the ordering of elements) are: 1. (1, 4), (2, 3) -> min(1, 4) + min(2, 3) = 1 + 2 = 3 2. (1, 3), (2, 4) -> min(1, 3) + min(2, 4) = 1 + 2 = 3 3. (1, 2), (3, 4) -> min(1, 2) + min(3, 4) = 1 + 3 = 4 So the maximum possible sum is 4.

Example 2:

Input: nums = [6,2,6,5,1,2] **Output:** 9 **Explanation:** The optimal pairing is (2, 1), (2, 5), (6, 6). min(2, 1) + min(2, 5) + min(6, 6) = 1 + 2 + 6 = 9.

Constraints:

* `1 <= n <= 104` * `nums.length == 2 * n` * `-104 <= nums[i] <= 104`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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