Problem 1877: Minimize Maximum Pair Sum in Array

Problem Information

Difficulty: Medium
Acceptance Rate: 81.60%
Paid Only: No
Tags: Array, Two Pointers, Greedy, Sorting

Problem Description

The **pair sum** of a pair `(a,b)` is equal to `a + b`. The **maximum pair sum** is the largest **pair sum** in a list of pairs.

* For example, if we have pairs `(1,5)`, `(2,3)`, and `(4,4)`, the **maximum pair sum** would be `max(1+5, 2+3, 4+4) = max(6, 5, 8) = 8`.

Given an array `nums` of **even** length `n`, pair up the elements of `nums` into `n / 2` pairs such that:

* Each element of `nums` is in **exactly one** pair, and * The **maximum pair sum** is **minimized**.

Return _the minimized**maximum pair sum** after optimally pairing up the elements_.

Example 1:

Input: nums = [3,5,2,3] **Output:** 7 **Explanation:** The elements can be paired up into pairs (3,3) and (5,2). The maximum pair sum is max(3+3, 5+2) = max(6, 7) = 7.

Example 2:

Input: nums = [3,5,4,2,4,6] **Output:** 8 **Explanation:** The elements can be paired up into pairs (3,5), (4,4), and (6,2). The maximum pair sum is max(3+5, 4+4, 6+2) = max(8, 8, 8) = 8.

Constraints:

* `n == nums.length` * `2 <= n <= 105` * `n` is **even**. * `1 <= nums[i] <= 105`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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