

# Problem 3727: Maximum Alternating Sum of Squares

## Problem Information

Difficulty: **Medium**

Acceptance Rate: 60.41%

Paid Only: No

Tags: Array, Greedy, Sorting

## Problem Description

You are given an integer array `nums`. You may **rearrange the elements** in any order.

The **alternating score** of an array `arr` is defined as:

$\text{score} = arr[0]^2 - arr[1]^2 + arr[2]^2 - arr[3]^2 + \dots$

Return an integer denoting the **maximum possible alternating score** of `nums` after rearranging its elements.

**Example 1:**

**Input:** `nums = [1,2,3]`

**Output:** 12

**Explanation:**

A possible rearrangement for `nums` is `[2,1,3]`, which gives the maximum alternating score among all possible rearrangements.

The alternating score is calculated as:

$\text{score} = 2^2 - 1^2 + 3^2 = 4 - 1 + 9 = 12$

**\*\*Example 2:\*\***

**\*\*Input:\*\*** nums = [1,-1,2,-2,3,-3]

**\*\*Output:\*\*** 16

**\*\*Explanation:\*\***

A possible rearrangement for `nums` is `[-3,-1,-2,1,3,2]`, which gives the maximum alternating score among all possible rearrangements.

The alternating score is calculated as:

$\text{score} = (-3)^2 - (-1)^2 + (-2)^2 - (1)^2 + (3)^2 - (2)^2 = 9 - 1 + 4 - 1 + 9 - 4 = 16$

**\*\*Constraints:\*\***

$1 \leq \text{nums.length} \leq 105$   $-4 \leq \text{nums}[i] \leq 4$

## Code Snippets

### C++:

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

    }
};
```

### Java:

```
class Solution {
    public long maxAlternatingSum(int[] nums) {

    }
}
```

### Python3:

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