Problem 3194: Minimum Average of Smallest and Largest Elements

Problem Information

Difficulty: Easy
Acceptance Rate: 85.11%
Paid Only: No
Tags: Array, Two Pointers, Sorting

Problem Description

You have an array of floating point numbers `averages` which is initially empty. You are given an array `nums` of `n` integers where `n` is even.

You repeat the following procedure `n / 2` times:

* Remove the **smallest** element, `minElement`, and the **largest** element `maxElement`, from `nums`. * Add `(minElement + maxElement) / 2` to `averages`.

Return the **minimum** element in `averages`.

Example 1:

Input: nums = [7,8,3,4,15,13,4,1]

Output: 5.5

Explanation:

step | nums | averages ---|---|--- 0 | [7,8,3,4,15,13,4,1] | [] 1 | [7,8,3,4,13,4] | [8] 2 | [7,8,4,4] | [8,8] 3 | [7,4] | [8,8,6] 4 | [] | [8,8,6,5.5] The smallest element of averages, 5.5, is returned.

Example 2:

Input: nums = [1,9,8,3,10,5]

Output: 5.5

Explanation:

step | nums | averages ---|---|--- 0 | [1,9,8,3,10,5] | [] 1 | [9,8,3,5] | [5.5] 2 | [8,5] | [5.5,6] 3 | [] | [5.5,6,6.5] **Example 3:**

Input: nums = [1,2,3,7,8,9]

Output: 5.0

Explanation:

step | nums | averages ---|---|--- 0 | [1,2,3,7,8,9] | [] 1 | [2,3,7,8] | [5] 2 | [3,7] | [5,5] 3 | [] | [5,5,5]

Constraints:

* `2 <= n == nums.length <= 50` * `n` is even. * `1 <= nums[i] <= 50`

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public double minimumAverage(int[] nums) {

}
}
```

Python3:

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