Problem 4: Median of Two Sorted Arrays

Problem Information

Difficulty: Hard
Acceptance Rate: 45.20%
Paid Only: No
Tags: Array, Binary Search, Divide and Conquer

Problem Description

Given two sorted arrays `nums1` and `nums2` of size `m` and `n` respectively, return **the median** of the two sorted arrays.

The overall run time complexity should be `O(log (m+n))`.

Example 1:

Input: nums1 = [1,3], nums2 = [2] **Output:** 2.00000 **Explanation:** merged array = [1,2,3] and median is 2.

Example 2:

Input: nums1 = [1,2], nums2 = [3,4] **Output:** 2.50000 **Explanation:** merged array = [1,2,3,4] and median is (2 + 3) / 2 = 2.5.

Constraints:

* `nums1.length == m` * `nums2.length == n` * `0 <= m <= 1000` * `0 <= n <= 1000` * `1 <= m + n <= 2000` * `-106 <= nums1[i], nums2[i] <= 106`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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