Problem 2422: Merge Operations to Turn Array Into a Palindrome

Problem Information

Difficulty: Medium
Acceptance Rate: 68.85%
Paid Only: Yes
Tags: Array, Two Pointers, Greedy

Problem Description

You are given an array `nums` consisting of **positive** integers.

You can perform the following operation on the array **any** number of times:

* Choose any two **adjacent** elements and **replace** them with their **sum**. * For example, if `nums = [1,_2,3_ ,1]`, you can apply one operation to make it `[1,5,1]`.

Return _the**minimum** number of operations needed to turn the array into a **palindrome**_.

Example 1:

Input: nums = [4,3,2,1,2,3,1] **Output:** 2 **Explanation:** We can turn the array into a palindrome in 2 operations as follows: - Apply the operation on the fourth and fifth element of the array, nums becomes equal to [4,3,2,**_3_** ,3,1]. - Apply the operation on the fifth and sixth element of the array, nums becomes equal to [4,3,2,3,**_4_**]. The array [4,3,2,3,4] is a palindrome. It can be shown that 2 is the minimum number of operations needed.

Example 2:

Input: nums = [1,2,3,4] **Output:** 3 **Explanation:** We do the operation 3 times in any position, we obtain the array [10] at the end which is a palindrome.

Constraints:

* `1 <= nums.length <= 105` * `1 <= nums[i] <= 106`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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