Problem 2009: Minimum Number of Operations to Make Array Continuous

Problem Information

Difficulty: Hard
Acceptance Rate: 52.16%
Paid Only: No
Tags: Array, Hash Table, Binary Search, Sliding Window

Problem Description

You are given an integer array `nums`. In one operation, you can replace **any** element in `nums` with **any** integer.

`nums` is considered **continuous** if both of the following conditions are fulfilled:

* All elements in `nums` are **unique**. * The difference between the **maximum** element and the **minimum** element in `nums` equals `nums.length - 1`.

For example, `nums = [4, 2, 5, 3]` is **continuous** , but `nums = [1, 2, 3, 5, 6]` is **not continuous**.

Return _the**minimum** number of operations to make _`nums` __**_continuous_**.

Example 1:

Input: nums = [4,2,5,3] **Output:** 0 **Explanation:** nums is already continuous.

Example 2:

Input: nums = [1,2,3,5,6] **Output:** 1 **Explanation:** One possible solution is to change the last element to 4. The resulting array is [1,2,3,5,4], which is continuous.

Example 3:

Input: nums = [1,10,100,1000] **Output:** 3 **Explanation:** One possible solution is to: - Change the second element to 2. - Change the third element to 3. - Change the fourth element to 4. The resulting array is [1,2,3,4], which is continuous.

Constraints:

* `1 <= nums.length <= 105` * `1 <= nums[i] <= 109`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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