Problem 1224: Maximum Equal Frequency

Problem Information

Difficulty: Hard
Acceptance Rate: 37.80%
Paid Only: No
Tags: Array, Hash Table

Problem Description

Given an array `nums` of positive integers, return the longest possible length of an array prefix of `nums`, such that it is possible to remove **exactly one** element from this prefix so that every number that has appeared in it will have the same number of occurrences.

If after removing one element there are no remaining elements, it's still considered that every appeared number has the same number of ocurrences (0).

Example 1:

Input: nums = [2,2,1,1,5,3,3,5] **Output:** 7 **Explanation:** For the subarray [2,2,1,1,5,3,3] of length 7, if we remove nums[4] = 5, we will get [2,2,1,1,3,3], so that each number will appear exactly twice.

Example 2:

Input: nums = [1,1,1,2,2,2,3,3,3,4,4,4,5] **Output:** 13

Constraints:

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

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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