Problem 2598: Smallest Missing Non-negative Integer After Operations

Problem Information

Difficulty: Medium
Acceptance Rate: 55.99%
Paid Only: No
Tags: Array, Hash Table, Math, Greedy

Problem Description

You are given a **0-indexed** integer array `nums` and an integer `value`.

In one operation, you can add or subtract `value` from any element of `nums`.

* For example, if `nums = [1,2,3]` and `value = 2`, you can choose to subtract `value` from `nums[0]` to make `nums = [-1,2,3]`.

The MEX (minimum excluded) of an array is the smallest missing **non- negative** integer in it.

* For example, the MEX of `[-1,2,3]` is `0` while the MEX of `[1,0,3]` is `2`.

Return _the maximum MEX of_`nums` _after applying the mentioned operation**any number of times**_.

Example 1:

Input: nums = [1,-10,7,13,6,8], value = 5 **Output:** 4 **Explanation:** One can achieve this result by applying the following operations: - Add value to nums[1] twice to make nums = [1,**_0_** ,7,13,6,8] - Subtract value from nums[2] once to make nums = [1,0,**_2_** ,13,6,8] - Subtract value from nums[3] twice to make nums = [1,0,2,**_3_** ,6,8] The MEX of nums is 4. It can be shown that 4 is the maximum MEX we can achieve.

Example 2:

Input: nums = [1,-10,7,13,6,8], value = 7 **Output:** 2 **Explanation:** One can achieve this result by applying the following operation: - subtract value from nums[2] once to make nums = [1,-10,_**0**_ ,13,6,8] The MEX of nums is 2. It can be shown that 2 is the maximum MEX we can achieve.

Constraints:

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

Code Snippets

C++:

```cpp
class Solution {
public:
    int findSmallestInteger(vector<int>& nums, int value) {

    }
};
```

Java:

```java
class Solution {
    public int findSmallestInteger(int[] nums, int value) {

    }
}
```

Python3:

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