

2598. Smallest Missing Non-negative Integer After Operations

Solved 

Medium

 Topics

 Companies

 Hint

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`

Python:

```
class Solution:
    def findSmallestInteger(self, nums: list[int], value: int) -> int:
        remainder_count = [0] * value

        for num in nums:
            rem = ((num % value) + value) % value
            remainder_count[rem] += 1

        result = 0
        while remainder_count[result % value] > 0:
            remainder_count[result % value] -= 1
            result += 1

        return result
```

JavaScript:

```
var findSmallestInteger = function(nums, value) {
    const remainderCount = new Array(value).fill(0);

    for (const num of nums) {
        const rem = ((num % value) + value) % value;
        remainderCount[rem]++;
    }

    let result = 0;
    while (remainderCount[result % value] > 0) {
        remainderCount[result % value]--;
        result++;
    }

    return result;
};
```

Java:

```
class Solution {
    public int findSmallestInteger(int[] nums, int value) {
        int[] remainderCount = new int[value];

        for (int num : nums) {
            int rem = ((num % value) + value) % value;
            remainderCount[rem]++;
        }
    }
}
```

```
int result = 0;
while (remainderCount[result % value] > 0) {
    remainderCount[result % value]--;
    result++;
}

return result;
}
```