

2357. Make Array Zero by Subtracting Equal Amounts

You are given a non-negative integer array `nums`. In one operation, you must:

- Choose a positive integer `x` such that `x` is less than or equal to the **smallest non-zero** element in `nums`.
- Subtract `x` from every **positive** element in `nums`.

Return the **minimum** number of operations to make every element in `nums` equal to `0`.

Example 1:

Input: `nums = [1,5,0,3,5]`

Output: 3

Explanation:

In the first operation, choose `x = 1`. Now, `nums = [0,4,0,2,4]`.

In the second operation, choose `x = 2`. Now, `nums = [0,2,0,0,2]`.

In the third operation, choose `x = 2`. Now, `nums = [0,0,0,0,0]`.

Example 2:

Input: `nums = [0]`

Output: 0

Explanation: Each element in `nums` is already 0 so no operations are needed.

Constraints:

- `1 <= nums.length <= 100`
- `0 <= nums[i] <= 100`

```
class Solution {
public:
    int minimumOperations(vector<int>& nums) {
        int n = nums.size(), minE = 0, result = 0;

        sort(nums.begin(), nums.end());

        for(int i=0; i<n; i++) {
            if(nums[i] > 0) {
                minE = nums[i];
                result++;

                for(int j=0; j<n; j++) {
                    nums[j] = nums[j] > 0 ? (nums[j] - minE) : 0;
                }
            }

            int flag = 1;
            for(auto num:nums) {
                if(num != 0) {
                    flag = 0;
                    break;
                }
            }
            if(flag) return result;
        }
        return result;
    }
};
```

#100daysofDSA



/rvislive

Rakesh Vishwakarma