

## 1658. Minimum Operations to Reduce X to Zero

Medium



1998



38



Add to List



Share

You are given an integer array `nums` and an integer `x`. In one operation, you can either remove the leftmost or the rightmost element from the array `nums` and subtract its value from `x`. Note that this **modifies** the array for future operations.

Return the **minimum number** of operations to reduce `x` to **exactly** `0` if it is possible, otherwise, return `-1`.

### Constraints:

- `1 <= nums.length <= 105`
- `1 <= nums[i] <= 104`
- `1 <= x <= 109`

### Example 1:

**Input:** `nums = [1,1,4,2,3], x = 5`

**Output:** `2`

**Explanation:** The optimal solution is to remove the last two elements to reduce `x` to zero.

### Example 2:

**Input:** `nums = [5,6,7,8,9], x = 4`

**Output:** `-1`

### Example 3:

**Input:** `nums = [3,2,20,1,1,3], x = 10`

**Output:** `5`

**Explanation:** The optimal solution is to remove the last three elements and the first two elements (5 operations in total) to reduce `x` to zero.

```
int minOperations(vector<int>& nums, int x) {
    int sum = accumulate(nums.begin(), nums.end(), 0);
    int l = 0, r = 0, res = INT_MAX, sz = nums.size();
    while (l <= r)
        if (sum >= x) {
            if (sum == x){
                res = min(res, l + sz - r);
            }

            if (r < sz) {
                sum = sum - nums[r++];
            } else {
                break;
            }
        } else {
            sum = sum + nums[l++];
        }

    return res == INT_MAX ? -1 : res;
}
```

# #100daysofDSA

---



/rvislive

**Rakesh Vishwakarma**