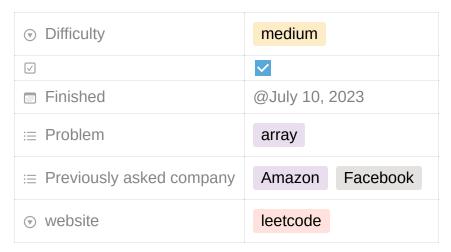
18. 4Sum



Question:

Given an array nums of n integers, return an array of all the **unique** quadruplets [nums[a], nums[b], nums[c], nums[d]] such that:

```
• 0 <= a, b, c, d < n
```

- a, b, c, and d are distinct.
- nums[a] + nums[b] + nums[c] + nums[d] == target

You may return the answer in any order.

Example 1:

```
Input: nums = [1,0,-1,0,-2,2], target = 0
Output: [[-2,-1,1,2],[-2,0,0,2],[-1,0,0,1]]
```

Example 2:

```
Input: nums = [2,2,2,2,2], target = 8
Output: [[2,2,2,2]]
```

Optimal solution:

Time complexity: O(n^3)

Space complextiy: O(1) or O(n)

18. 4Sum 1

```
l, r = start, len(nums)-1
while l < r:
    if nums[l] + nums[r] < target:
        l += 1
    elif nums[l] + nums[r] > target:
        r -= 1
    else:
        res.append(quad + [nums[l], nums[r]])
        l += 1
        while l < r and nums[l] == nums[l-1]:
        l += 1
kSum(4, 0, target)
return res</pre>
```

18. 4Sum 2