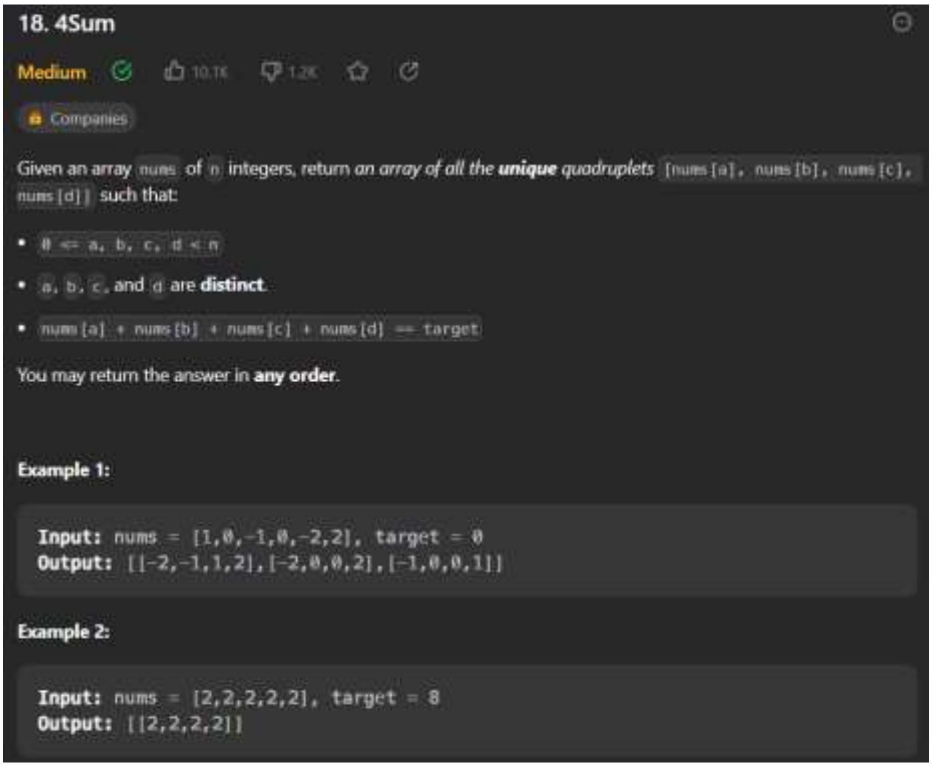
LEETCODE SOLUTIONS - YOGESH MUNEES



18. 4Sum

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**.

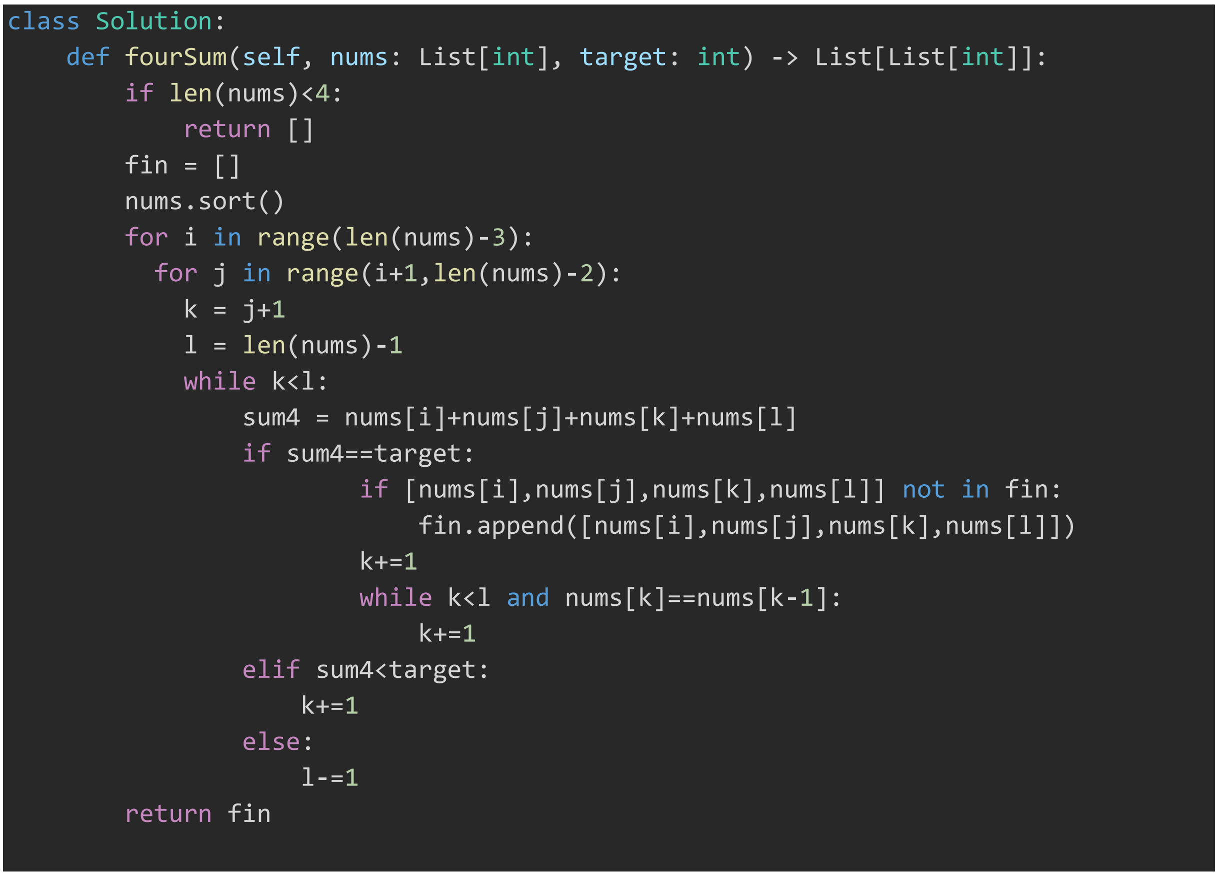
Example 1:

```
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]]
```

Solution:



```
class Solution:
    def fourSum(self, nums: List[int], target: int) -> List[List[int]]:
        if len(nums)<4:
            return []
        fin = []
        nums.sort()
        for i in range(len(nums)-3):
          for j in range(i+1,len(nums)-2):
            k = j+1
            l = len(nums)-1
            while k<l:
                sum4 = nums[i]+nums[j]+nums[k]+nums[l]
                if sum4==target:
                        if [nums[i],nums[j],nums[k],nums[l]] not in fin:
                            fin.append([nums[i],nums[j],nums[k],nums[l]])
                        k+=1
                        while k<l and nums[k]==nums[k-1]:
                            k+=1
                elif sum4<target:
                    k+=1
                else:
                    l-=1
        return fin
```