4 Sum

Question: Given an array S of n integers, are there elements a, b, c, and d in S such that a + b + c + d = target?

Find all unique quadruplets in the array which gives the sum of target.

Note:

Elements in a quadruplet (a,b,c,d) must be in non-descending order. (ie, a ⩽ b ⩽ c ⩽ d)

The solution set must not contain duplicate quadruplets.

For example, given array S = {1 0 -1 0 -2 2}, and target = 0.

A solution set is: (-1, 0, 0, 1); (-2, -1, 1, 2); (-2, 0, 0, 2)

Solutions:

```
class Solution:
    def fourSum(self, nums, target):
        answer = []
        nums.sort()
        length = len(nums)
        for k in range(length-3):
            if nums[k]+nums[k+1]+nums[k+2]+nums[k+3] > target:
                  break
            for i in range(k+1,length-2):
                low = i+1
                high = length - 1
                while(low < high):
                    temp = nums[i]+nums[low]+nums[high]+nums[k]
                    if temp == target:
                        ans = [nums[i],nums[low],nums[high],nums[k]]
                        ans.sort()
                        low = low + 1
                        high = high - 1
                        if ans not in answer:
                            answer.append(ans)
                        while low < high and nums[high+1] == nums[high]:  ##speed up, jump the same value
                            high -= 1
                        while low < high and nums[low] == nums[low-1]:
                            low += 1
                    elif temp > target:
                        high = high -1
                    else:
                        low = low + 1
        return answer
Solution().fourSum([1, 0, -1, 0, -2, 2], 0)
```