18. 4Sum

```
class Solution:
    def fourSum(self, nums: List[int], target: int) -> List[List[int]]:
        if len(nums)<4:
            return []
        nums.sort()
        n = len(nums)
        res = []
        for i in range(0,n):
            if i>0 and nums[i]==nums[i-1]:
                continue
            else:
                tempTarget = target-nums[i]
                temp = self.threeSum(nums,i+1,n-1,tempTarget)
                if len(temp)>0:
                    temp = self.result(nums[i],temp)
            res = res+temp
        return res

    def threeSum(self,nums,start,end,target):
        res = []
        # n = len(nums)
        for i in range(start,end+1):
            if i>start and nums[i]==nums[i-1]:
                continue
            else:
                tempTarget = target-nums[i]
                temp = self.twoSums(nums,i+1,end,tempTarget)
                if len(temp)>0:
                    temp = self.result(nums[i],temp)
            res = res+temp
        return res

    def result(self,ele,ans):
        for i in range(len(ans)):
            ans[i] = ans[i]+[ele]
        return ans

    def twoSums(self,nums,start,end,target):
        ans = []
```

```
        while start<end:
            temp = nums[start]+nums[end]
            if temp == target:
                ans.append([nums[start],nums[end]])
                start+=1
                end-=1

                while start<end and nums[start]==nums[start-1]:
                    start+=1
                while start<end and nums[end]==nums[end+1]:
                    end-=1
            elif temp>target:
                end-=1
            else:
                start+=1

        return ans
```