1. 4Sum

Medium

Given an array nums of *n* integers and an integer target, are there elements *a*, *b*, *c*, and *d* in nums such that *a* + *b* + *c* + *d* = target? Find all unique quadruplets in the array which gives the sum of target.

**Note:**

The solution set must not contain duplicate quadruplets.

**Example:**

Given array nums = [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]  
]

**分析：思路和3sum相同，注意里面去重的方法**

class Solution {  
public:  
 vector<vector<int>> fourSum(vector<int>& nums, int target) {  
 vector<vector<int> >ans;  
 int n = nums.size();  
 sort(nums.begin(), nums.end());  
 for(int i = 0; i < n; ++i){  
 if(i > 0 && nums[i] == nums[i - 1])continue;  
 for(int j = i + 1; j < n; ++j){  
 if(j > i + 1 && nums[j] == nums[j - 1])continue;  
 int m = target - (nums[i] + nums[j]);  
 int p = j + 1, q = n - 1;  
 while(p < q){  
 if(nums[p] + nums[q] == m){  
 ans.push\_back({nums[i], nums[j], nums[p], nums[q]});  
 while(p + 1 < q && nums[p + 1] == nums[p])p++;  
 while(q - 1 > p && nums[q - 1] == nums[q])q--;  
 p++;  
 q--;  
 }else if(nums[p] + nums[q] < m)p++;  
 else q--;  
 }  
 }  
 }  
 return ans;  
 }  
};