/\*

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: 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]

]

题目：4sum=target

思路：两重for固定两个数，两个指针移动确定后两个数

\*/

class Solution {

public:

vector<vector<int>> fourSum(vector<int>& nums, int target)

{

vector<vector<int>> result;

if(nums.size()<4)

return result;

vector<int> m1;

sort(nums.begin(),nums.end());

int left,right;

for(int i=0;i<nums.size()-3;i++)

{

if(nums[i]==nums[i-1] && i>0)

continue;

for(int j=i+1;j<nums.size()-2;j++)

{

if(nums[j]==nums[j-1] && j>i+1)

continue;

left=j+1;

right=nums.size()-1;

while(left<right)

{

if(nums[i]+nums[j]+nums[left]+nums[right]>target)

right--;

else if(nums[i]+nums[j]+nums[left]+nums[right]<target)

left++;

else

{

m1.push\_back(nums[i]);

m1.push\_back(nums[j]);

m1.push\_back(nums[left]);

m1.push\_back(nums[right]);

if(result.size()>0)

{

if(m1!=result[result.size()-1])

result.push\_back(m1);

}

else

result.push\_back(m1);

m1.clear();

right--;

left++;

}

}

}

}

return result;

}

};