/\*

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

way-1：两重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)

{

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

vector<vector<int> > ret;

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

{

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

continue;

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

{

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

continue;

int key = target - (nums[i] + nums[j]);

int l = j + 1, r = nums.size() - 1;

while (l < r)

{

if (nums[l] + nums[r] == key)

{

vector<int> temp;

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

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

temp.push\_back(nums[l]);

temp.push\_back(nums[r]);

if (ret.size() == 0 || temp != ret[ret.size()-1])

ret.push\_back(temp);

l++;

r--;

}

else if (nums[l] + nums[r] < key)

l++;

else

r--;

}

}

}

return ret;

}

};