只需要找出一对sum值为target的即可。

第一次提交时候，我的代码是这样的：

class Solution {

public:

vector<int> twoSum(vector<int>& nums, int target) {

int length=nums.size();

vector<int> output(2);

if(length<2)

return output;

for(int i=0;i<length;i++)

{

int first=nums[i];

int second=target-first;

for(int j=i+1;j<length;j++)

{

if(second==nums[j])

{

output[0]=i+1;

output[1]=j+1;

return output;

}

}

}

}

};

当时，由于vector不了解，还自己写了测试用例，因此画了三个小时完成，但是提交的时候，部分案例无法通过。观察这些案例，发现都是nums的size很大的那种，于是明白，自己代码思路太简单，时间复杂度太长。

第二次查看Discussion别人的代码，就知道优化方式是：先数组进行一次排序，再进行二分查找。

class Solution {

public:

vector<int> twoSum(vector<int>& nums, int target) {

vector<int> copy;

int length=nums.size();

copy=nums;

vector<int> output(2);

vector<int>::iterator find1,find2;

if(length<2)

return output;

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

int i=0;

int j=length-1;

int first=0,second=0;

while(i<j)

{

if(nums[i]+nums[j]==target)

break;

else if(nums[i]+nums[j]>target)

j--;

else

i++;

}

if(nums[i]+nums[j]==target)

{

for(int m=0;m<length;m++)

{

if(copy[m]==nums[i])

first=m;

if(copy[m]==nums[j])

second=m;

}

if(first<second)

{

output[0]=first;

output[1]=second;

}

else

{

output[0]=second;

output[1]=first;

}

return output;

}

else

return output;

}

};

第二次提交没注意小细节，即输出结果是index+1；同时有些案例无法通过[0,1,4,0], 0；我的输出是[4,4],实际预期输出应该是[1,4],显然我没有考虑数组中存在相同数字的情况。

最终，快速的修改之后，代码是这样的：

class Solution {

public:

vector<int> twoSum(vector<int>& nums, int target) {

vector<int> copy;

int length=nums.size();

copy=nums;

vector<int> output(2);

vector<int>::iterator find1,find2;

if(length<2)

return output;

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

int i=0;

int j=length-1;

int first=0,second=0;

while(i<j)

{

if(nums[i]+nums[j]==target)

break;

else if(nums[i]+nums[j]>target)

j--;

else

i++;

}

if(nums[i]+nums[j]==target)

{

for(int m=0;m<length;m++)

{

if(copy[m]==nums[i])

first=m;

}

for(int m=length-1;m>=0;m--)

{

if(copy[m]==nums[j])

second=m;

}

if(first<second)

{

output[0]=first+1;

output[1]=second+1;

}

else

{

output[0]=second+1;

output[1]=first+1;

}

return output;

}

else

return output;

}

};

结果是Accepted，虽然问题不难，但是对于我来历时太久，因为缺乏锻炼，但是看到accepted还是超开心。恭喜第一步迈出。