# 哈希结构

## 哈希建表（map函数）

【两数之和】

两遍哈希->考虑一遍哈希？

class Solution {

public:

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

map<int,int> a;//建立hash表存放数组元素

vector<int> b(2,-1);//存放结果

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

a.insert(map<int,int>::value\_type(nums[i],i));//第一次将所有的键值对创建为一个哈希表。

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

{

if(a.count(target-nums[i])>0&&(a[target-nums[i]]!=i))

//判断是否找到目标元素且目标元素不能是本身（必须要有两个nums[i]中的“3”）

//count()函数返回map中键出现的次数（0/1）

{

b[0]=i;

b[1]=a[target-nums[i]];

break;

}

}

return b;

};

};

class Solution {

public:

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

map<int,int> a;//提供一对一的hash

vector<int> b(2,-1);//用来承载结果，初始化一个大小为2，值为-1的容器b

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

{

if(a.count(target-nums[i])>0)

{

b[0]=a[target-nums[i]];

b[1]=i;

break;

}

a[nums[i]]=i;//反过来放入map中，用来获取结果下标

}

return b;

};

};