**描述**

给一个整数数组，找到两个数使得他们的和等于一个给定的数 *target*。

你需要实现的函数twoSum需要返回这两个数的下标, 并且第一个下标小于第二个下标。注意这里下标的范围是 0 到 *n-1*。

这道题因为空间复杂度，所以就比较麻烦。我是直接上手用了哈希表，因为既涉及到下标，又涉及到相应的数字。我最后就用了哈希表了。整个的思路很简单。就是先创建一个哈希表，把对应的数字和下标填进去。

然后下一步很简单。就是直接查找。查找target-number[i]是否在这个哈希表里，如果是，就返回下标。

代码如下：

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

// write your code here

unordered\_map<int,int>m;//第一步，创建一个hash表

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

m[numbers[i]]=i;

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

if(m.find(target-numbers[i])!=m.end()&&m[target-numbers[i]]!=i)

return{

i,m[target-numbers[i]]

};

}

return {};

}

