

剑指Offer（四十二）：和为S的两个数字

🕒 2018年1月18日 12:56:24 📄 3 🌡 2,516 °C 🛠 编辑



一、前言

本系列文章为《剑指Offer》刷题笔记。

刷题平台：[牛客网](#)

书籍下载：[共享资源](#)

二、题目

输入一个递增排序的数组和一个数字S，在数组中查找两个数，是的他们的和正好是S，如果有多对数字的和等于S，输出两个数的乘积最小的。

输出描述：

对应每个测试案例，输出两个数，小的先输出。

1、思路

对于一个数组，我们可以定义两个指针，一个从左往右遍历（pleft），另一个从右往左遍历（pright）。首先，我们比较第一个数字和最后一个数字的和curSum与给定数字sum，如果curSum < sum，那么我们就加大输入值，所以，pleft向右移动一位，重复之前的计算；如果curSum > sum，那么我们就减小输入值，所以，pright向左移动一位，重复之前的计算；如果相等，那么这两个数字就是我们要找的数字，直接输出即可。

这么做的好处是，也保证了乘积最小。

2、代码

C++:


C++

```
1 class Solution {
2 public:
3     vector<int> FindNumbersWithSum(vector<int> array,int sum) {
4         vector<int> result;
5         int length = array.size();
6         if(length < 1){
7             return result;
8         }
9         int pright = length - 1;
10        int pleft = 0;
11
12        while(pright > pleft){
13            int curSum = array[pleft] + array[pright];
14            if(curSum == sum){
15                result.push_back(array[pleft]);
16                result.push_back(array[pright]);
17                break;
18            }
19            else if(curSum < sum){
20                pleft++;
21            }
22            else{
23                pright--;
24            }
25        }
26        return result;
27    }
28 };
```

Python:

Pytho

```
1 # -*- coding:utf-8 -*-
2 class Solution:
3     def FindNumbersWithSum(self, array, tsum):
4         # write code here
5         if len(array) <= 1:
6             return []
7         left, right = 0, len(array)-1
8         while left < right:
9             if array[left] + array[right] == tsum:
10                 return array[left], array[right]
11             elif array[left] + array[right] < tsum:
12                 left += 1
13             else:
14                 right -= 1
15         return []
```



微信公众号

分享技术，乐享生活：微信公众号搜索「JackCui-AI」关注一个在互联网摸爬滚打的潜行者。

猛兽总是独行，牛羊才成群结队。--- 鲁迅