

剑指Offer（四十一）： 和为S的连续正数序列

🕒 2018年1月18日 10:24:49 📄 2 🌡 2,916 °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

小明很喜欢数学,有一天他在做数学作业时,要求计算出9~16的和,他马上就写出了正确答案是100。但是他并不满足于此,他在想究竟有多少种连续的正数序列的和为100(至少包括两个数)。没多久,他就得到另一组连续正数和为100的序列:18,19,20,21,22。现在把问题交给你,你能不能也很快地找出所有和为S的连续正数序列? Good Luck!

输出描述：

输出所有和为S的连续正数序列。序列内按照从小至大的顺序，序列间按照开始数字从小到大的顺序。

1、思路

这道题还是蛮简单的。

设定两个指针，一个指向第一个数，一个指向最后一个数，在此之前需要设定第一个数和最后一个数的值，由于是正数序列，所以可以把第一个数设为1，最后一个数为2（因为要求是连续正数序列，最后不可能和第一个数重合）。下一步就是不断改变第一个数和最后一个数的值，如果从第一个数到最后一个数的和刚好是要求的和，那么把所有的数都添加到一个序列中；如果大于要求的和，则说明从第一个数到最后一个数之间的范围太大，因此减小范围，需要把第一个数的值加1，同时把当前和减去原来的第一个数的值；如果小于要求的和，说明范围太小，因此把最后一个数加1，同时把当前的和加上改变后的最后一个数的值。这样，不断修改第一个数和最后一个数的值，就能确定所有连续正数序列的和等于S的序列了。

注意：初中的求和公式应该记得吧，首项加尾项的和乘以个数除以2，即 $sum = (a + b) * n / 2$ 。

2、代码

C++:


```
1 class Solution {
2 public:
3     vector<vector<int> > FindContinuousSequence(int sum) {
4         vector<vector<int> > result;
5         // 高位指针和低位指针
6         int phigh = 2, plow = 1;
7
8         // 终止条件是phigh等于sum
9         while(phigh < sum){
10             // 当前和，使用求和公式s = (a+b) * n / 2
11             int curSum = (plow + phigh) * (phigh - plow + 1) >> 1;
12             if(curSum < sum){
13                 phigh++;
14             }
15             if(curSum == sum){
16                 vector<int> temp;
17                 for(int i = plow; i <= phigh; i++){
18                     temp.push_back(i);
19                 }
20                 result.push_back(temp);
21                 plow++;
22             }
23         }
24         return result;
25     }
26 }
```

```
20         result.push_back(temp);
21         plow++;
22     }
23     if(curSum > sum){
24         plow++;
25     }
26 }
27 return result;
28 }
29 };
```

Python:

Pytho

```
1 # -*- coding:utf-8 -*-
2 class Solution:
3     def FindContinuousSequence(self, tsum):
4         # write code here
5         result = []
6         low, high = 1, 2
7         while low < high:
8             curSum = (low + high) * (high - low + 1) / 2
9             if curSum == tsum:
10                 tmp = []
11                 for i in range(low, high+1):
12                     tmp.append(i)
13                 result.append(tmp)
14                 low += 1
15             elif curSum < tsum:
16                 high += 1
17             else:
18                 low += 1
19         return result
```



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。

不在沉默中爆发，就在沉默中灭亡。--- 鲁迅