

题目一 和为s的两个数字

面试题 57：和为 s 的数字

题目一：和为 s 的两个数字。

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，使得它们的和正好是 s 。如果有多对数字的和等于 s ，则输出任意一对即可。

解

双指针，一个在头，一个在尾

```
vector<int> find(vector<int> num,int s)
{
    vector<int> res;
    if(num.empty())
        return res;
    int i=0,j=num.size()-1;
    while(i<j)
    {
        if(num[i]+num[j] == s)
        {
            res.push_back(num[i]);
            res.push_back(num[j]);
        }
        else if(num[i]+num[j] > s)
            j--;
        else
            i++;
    }
    return res;
}
```

题目二 和为s的连续正数序列

题目二：和为 s 的连续正数序列。

输入一个正数 s ，打印出所有和为 s 的连续正数序列（至少含有两个数）。例如，输入 15，由于 $1+2+3+4+5=4+5+6=7+8=15$ ，所以打印出 3 个连续序列 1~5、4~6 和 7~8。

解

滑动窗口，或者也叫双指针。两个指针都只向右走。

有了解决前面问题的经验，我们也考虑用两个数 **small** 和 **big** 分别表示序列的最小值和最大值。首先把 **small** 初始化为 1，**big** 初始化为 2。如果从 **small** 到 **big** 的序列的和大于 s ，则可以从序列中去掉较小的值，也就是增大 **small** 的值。如果从 **small** 到 **big** 的序列的和小于 s ，则可以增大 **big**，让这个序列包含更多的数字。因为这个序列至少要有两个数字，我们一直增加 **small** 到 $(1+s)/2$ 为止。

```
vector<vector<int>> findSequence(int s)
{
    vector<vector<int>> res;
    if(s<=0)
        return res;
    int left=1,right=2;
    int a=(1+s)/2;
    while(left<a)
    {
        int cur=(right+left) * (right-left+1)/2;
        if(cur<s)
            right++;
        else if(cur==s)
        {
            vector<int> tmp;
            for(int i=left;i<=right;i++)
                tmp.push_back(i);
            res.push_back(tmp);
            left++;
        }
        else
            left++;
    }
    return res;
}
```