和为S的数字和和为S连续正数序列

# 和为S的两个数字

## 题目介绍

**题目描述**：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。

## 思路分析

\* 思路分析：**start和end两个索引**，初始分别指向数组开头和末尾；

* + **若两者之和等于S**，则计算两者乘积，若乘积小于上次的乘积，则保存到result中；一定要**start++或end--**；(均可以)
    - **若两者之和小于S**，则start++；
    - **若两者之和大于S**，则end--。

## Java代码

public ArrayList<Integer> FindNumbersWithSum(int [] nums, int S) {

ArrayList<Integer> result = new ArrayList<Integer>();

if(nums==null||nums.length==0) return result;

int start = 0,end = nums.length-1;

int currentProduct = Integer.MAX\_VALUE;

while(start < end){

int currentSum = nums[start] + nums[end];

if(currentSum == S){

int product = nums[start]\*nums[end];

if(product < currentProduct){

**currentProduct = product;//容易忽略**

result.add(0,nums[start]);

result.add(1,nums[end]);

}

**end--;//或者start++**

}else if(currentSum < S){

start++;

}else{

end--;

}

}

return result;

}

测试代码：

@Test

public void test(){

int[] nums = {2,3,4,5,6,6,7,8,9,10,12,18};

ArrayList<Integer> result = FindNumbersWithSum(nums,12);

System.out.println(result.toString());

}

输出:[6,6]

# 剑指Offer57:和为S的连续正数序列

## 题目介绍：

\* 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,你能不能也很快的找出所有和为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

## 思路分析

\* 思路分析： 首先两个正数最小为3，则S小于3则返回空list；至少是连个正数，则N-1+N<=S;则N<=(S+1)/2,循环终止条件就是right大于(N+!)/2

\* 定义两个变量left和right，伸缩移动，有个sum保存left到right的和，

\* 若sum小于S，则right向前移动，同时sum加上新的right值；

\* 若sum大于S，则left向前移动，同时sum减去旧的left值；

\* 若sum等于S，则left至right就是一个结果，存到结果集合中；然后right就行移动一位，left向前移动两位。

## Java代码

public ArrayList<ArrayList<Integer>> FindContinuousSequence(int S) {

ArrayList<ArrayList<Integer>> results = new ArrayList<ArrayList<Integer>>();

if(S < 3) return results;//连个正数最小为3

int left = 1,right = 2,sum = 3;

int maxRight = (S+1)>>1;

while(**right <= maxRight**){//&&left<right是多余的，因为任何小于maxRight的连续两个数之和肯定小于S，

// 下一步肯定是right右移,不可能让left和right重合更不可能left超过right

if(sum < S){

sum += ++right;//加上新的right值

}else if(sum > S){

sum -= left++;//减去旧的left值

}else{

ArrayList<Integer> result = new ArrayList<Integer>();

for(int i = left;i <= right;i++)

result.add(i);

results.add(result);

sum += ++right;

sum -= left++;

sum -= left++;

}

}

return results;

}

测试：

@Test  
public void test(){  
 ArrayList<ArrayList<Integer>> results = FindContinuousSequence(100);  
 for(ArrayList<Integer> result : results)  
 System.*out*.println("result = " + result);  
}

result = [9, 10, 11, 12, 13, 14, 15, 16]

result = [18, 19, 20, 21, 22]