和为S的数字

# 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]