

给定一个含有  $n$  个正整数的数组和一个正整数  $s$ ，找出该数组中满足其和  $\geq s$  的长度最小的连续子数组。如果不存在符合条件的连续子数组，返回 0。

示例：

输入： $s = 7$ ,  $nums = [2, 3, 1, 2, 4, 3]$

输出：2

解释：子数组  $[4, 3]$  是该条件下的长度最小的连续子数组。

进阶：

如果你已经完成了  $O(n)$  时间复杂度的解法，请尝试  $O(n \log n)$  时间复杂度的解法。

思路，用双指针进行数组的遍历，当当前和小于 $s$ 时，右边指针移动，和增加，当大于 $s$ 时，统计长度，有点指针增加，和减少，用此动态的遍历得到答案。

代码：

```
class Solution {
public:
    int minSubArrayLen(int s, vector<int>& nums) {
        int i=0, j=0;
        int sum=0;
        int min=0;
        while(j<nums.size()){
            sum+=nums[j];
            j++;
            while(sum>=s){
                sum-=nums[i];
                if(min==0){
                    min=j-i;
                }else if(j-i<min){
                    min=j-i;
                }
                i++;
            }
        }
        return min;
    }
};
```