

Python-数组滑窗-有 N 个正整数组成的一个序列

题目描述:

有 N 个正整数组成的一个序列。给定整数 sum ，求长度最长的连续子序列，使他们的和等于 sum ，返回此子序列的长度，如果没有满足要求的序列，返回 -1 。

输入描述:

序列: 1,2,3,4,2

sum : 6

输出描述:

序列长度: 3

补充说明:

输入序列仅由数字和英文逗号构成，数字之间采用英文逗号分隔；

序列长度: $1 \leq N \leq 200$;

输入序列不考虑异常情况，由题目保证输入序列满足要求。

示例 1

输入:

1,2,3,4,2

6

输出:

3

说明:

解释: 1,2,3 和 4,2 两个序列均能满足要求，所以最长的连续序列为 1,2,3，因此结果为

3

示例 2

输入:

1,2,3,4,2

20

输出:

-1

说明:

解释: 没有满足要求的子序列, 返回 -1

输入获取

```
arr = list(map(int, input().split(",")))
```

```
sumV = int(input())
```

算法入口

```
def getResult():
```

```
    ans = -1
```

```
    if sumV <= 0:
```

```
        return ans
```

```
    l = 0
```

```
    r = 0
```

```
    n = len(arr)
```

```
    total = arr[l]
```

```
while True:
```

```
    if total > sumV:
```

```
        l += 1
```

```
        total -= arr[l - 1]
```

```
        if r < l < n:
```

```
            r = l
```

```
            total += arr[r]
```

```
        elif total < sumV:
```

```
            r += 1
```

```
            if r < n:
```

```
                total += arr[r]
```

```
        else:
```

```
            break
```

```
    else:
```

```
        ans = max(ans, r - l + 1)
```

```
        l += 1
```

```
        r += 1
```

```
        if r < n:
```

```
            total += arr[r] - arr[l - 1]
```

```
        else:
```

```
            break
```

```
return ans
```

```
# 算法调用
```

```
print(getResult())
```