

华为OD机试 - 阿里巴巴找黄金宝箱(V) (Java & JS & Python)

原创

伏城之外

已于 2023-05-30 15:54:04 修改

1638

收藏 4

版权

分类专栏:

华为OD机试AB (Java & JS & Python)

文章标签:

算法

华为机试

Java

JavaScript

Python

OD

华为OD机试AB (Ja...
该专栏为热销专栏榜 第2名

同时被 2 个专栏收录

¥59.90
¥99.00

3381 订阅

371 篇文章

已订阅

题目描述

一贫如洗的樵夫阿里巴巴在去砍柴的路上，无意中发现了强盗集团的藏宝地，藏宝地有编号从0~N的箱子，每个箱子上面贴有一个数字。阿里巴巴念出一个咒语数字k(k<N)，找出连续k个宝箱数字和的最大值，并输出该最大值。

输入描述

第一行输入一个数字字符串，数字之间使用逗号分隔，例如：2,10,-3,-8,40,5

- 1 ≤ 字符串中数字的个数 ≤ 100000
- 10000 ≤ 每个数字 ≤ 10000

第二行输入咒语数字，例如：4，咒语数字大小小于宝箱的个数

输出描述

连续k个宝箱数字和的最大值，例如：39

用例

输入	2,10,-3,-8,40,5 4
输出	39
说明	无

输入	8 1
输出	8
说明	无

题目解析

本题可以使用 **滑动窗口** 解决。

题目说：

找出连续 k 个宝箱数字和的最大值

其中连续 k 个刚好可以用一个固定长度为 k 的滑动窗口去限定。

我们只需要花费 $O(k)$ 的时间求解出初始滑动窗口的数字和，之后每右移一个后的新状态滑窗的数字和都可以依赖于前一个滑窗的数字和求得：

初始滑窗	2	10	-3	-8	40	5	
	2	10	-3	-8	40	5	
	2	10	-3	-8	40	5	

如上图所示，除了初始滑窗，所有的滑窗的数字和 $cur = \text{之前滑窗的数字和} - \text{红色失去的数字} + \text{绿色新增的数字}$

```
1 import java.util.Arrays;
2 import java.util.Scanner;
3
4 public class Main {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         int[] arr =
9             Arrays.stream(sc.nextLine().split(",")).mapToInt(Integer::parseInt).toArray();
10
11         int k = Integer.parseInt(sc.nextLine());
12
13         System.out.println(getResult(arr, k));
14     }
15
16     public static int getResult(int[] arr, int k) {
17         int window_sum = 0;
18
19         for (int i = 0; i < k; i++) {
20             window_sum += arr[i];
21         }
22     }
23 }
```

```
20
21     int ans = window_sum;
22
23     for (int i = 1; i <= arr.length - k; i++) {
24         window_sum -= arr[i - 1];
25         window_sum += arr[i + k - 1];
26         ans = Math.max(ans, window_sum);
27     }
28
29     return ans;
30 }
31 }
```