

【华为OD机考 统一考试机试C卷】整数对最小和 (C++ Java JavaScript Python C语言)

华为OD机考:统一考试 C卷 + D卷 + B卷 +A卷

目前在考C卷，经过两个月的收集整理，**C卷真题已基本整理完毕**

抽到原题的概率为2/3到3/3，**也就是最少抽到两道原题**。请注意：大家刷完C卷真题，最好要把B卷的真题刷一下，因为C卷的部分真题来自B卷。

另外订阅专栏还可以联系笔者开通在线 **OJ** 进行刷题，提高刷题效率。

真题目录：华为OD机考机试 真题目录 (C卷 + D卷 + B卷 + A卷) + 考点说明

专栏：2023华为OD机试(B卷+C卷+D卷) (C++JavaJSPy)

华为OD面试真题精选：华为OD面试真题精选

在线OJ：点击立即刷题，模拟真实机考环境

题目描述：整数对最小和（分值100）

给定两个整数数组array1、array2，数组元素按升序排列。

假设从array1、array2中分别取出一个元素可构成一对元素，现在需要取出k对元素，

并对取出的所有元素求和，计算和的最小值。

注意：

两对元素如果对应于array1、array2中的两个下标均相同，则视为同一对元素。

输入描述

输入两行数组array1、array2，每行首个数字为数组大小size(0 < size <= 100);

$0 < \text{array1}[i] \leq 1000$

$0 < \text{array2}[i] \leq 1000$

接下来一行为正整数k

0 < k <= array1.size() * array2.size()

输出描述

满足要求的最小和

用例

输入	3 1 1 2 3 1 2 3 2
输出	4
说明	用例中，需要取2对元素 取第一个数组第0个元素与第二个数组第0个元素组成1对元素[1,1]; 取第一个数组第1个元素与第二个数组第0个元素组成1对元素[1,1]; 求和为1+1+1+1=4，为满足要求的最小和。

C++

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  int main() {
6      // 输入数组array1的大小
7      int size1;
8      std::cin >> size1;
9
10     // 创建大小为size1的vector来存储array1的元素
11     std::vector<int> array1(size1);
12     for (int i = 0; i < size1; i++) {
13         int x;
```

```

13     // 逐个读取array1的元素并存储到vector中
14     std::cin >> array1[i];
15 }
16
17 // 输入数组array2的大小
18 int size2;
19 std::cin >> size2;
20
21 // 创建大小为size2的vector来存储array2的元素
22 std::vector<int> array2(size2);
23 for (int i = 0; i < size2; i++) {
24     // 逐个读取array2的元素并存储到vector中
25     std::cin >> array2[i];
26 }
27
28 // 输入k的值
29 int k;
30 std::cin >> k;
31
32 // 创建一个vector来存储所有可能的元素对的和
33 std::vector<int> pairsSum;
34 for (int value1 : array1) {
35     for (int value2 : array2) {
36         // 将array1和array2中的元素两两相加, 并将结果存储到pairsSum中
37         pairsSum.push_back(value1 + value2);
38     }
39 }
40
41 // 对pairsSum中的元素进行排序
42 std::sort(pairsSum.begin(), pairsSum.end());
43
44 // 计算前k个元素的和
45 int minSum = 0;
46 for (int i = 0; i < k; i++) {
47     minSum += pairsSum[i];
48 }
49
50 // 输出最小和
51 std::cout << minSum << std::endl;
52
53

```

```
54 |     return 0;
55 | }
```

JavaScript

```
1 | const readline = require('readline');
2 |
3 | // 创建readline接口实例
4 | const rl = readline.createInterface({
5 |     input: process.stdin,
6 |     output: process.stdout
7 | });
8 |
9 | rl.on('line', (array1Input) => {
10 |     // 将输入的字符串按空格分割为数组, 并将每个元素转换为数字, 然后去除第一个元素
11 |     const array1 = array1Input.split(' ').map(Number).slice(1);
12 |
13 |     rl.on('line', (array2Input) => {
14 |         // 将输入的字符串按空格分割为数组, 并将每个元素转换为数字, 然后去除第一个元素
15 |         const array2 = array2Input.split(' ').map(Number).slice(1);
16 |
17 |         rl.on('line', (kInput) => {
18 |             // 将输入的字符串转换为整数
19 |             const k = parseInt(kInput);
20 |
21 |             // 创建一个空数组pairsSum
22 |             const pairsSum = [];
23 |
24 |             // 嵌套循环, 将array1和array2中的元素两两相加, 并将结果存储到pairsSum中
25 |             for (const value1 of array1) {
26 |                 for (const value2 of array2) {
27 |                     pairsSum.push(value1 + value2);
28 |                 }
29 |             }
30 |
31 |             // 对pairsSum中的元素进行排序
32 |             pairsSum.sort();
33 |
34 |             // 取出pairsSum中前k个元素, 并使用reduce方法计算它们的和
35 | }
```

```

35     const minSum = pairsSum.slice(0, k).reduce((sum, value) => sum + value, 0);
36
37     // 输出最小和
38     console.log(minSum);
39
40     // 关闭readLine接口, 结束程序的执行
41     rl.close();
42   });
43 });
44 });
45

```

Java

```

1  import java.util.ArrayList;
2  import java.util.Collections;
3  import java.util.List;
4  import java.util.Scanner;
5
6  public class Main {
7      public static void main(String[] args) {
8          Scanner scanner = new Scanner(System.in);
9
10         // 输入第一个数组
11         int size1 = scanner.nextInt();
12         List<Integer> array1 = new ArrayList<>();
13         for (int i = 0; i < size1; i++) {
14             array1.add(scanner.nextInt());
15         }
16
17         // 输入第二个数组
18         int size2 = scanner.nextInt();
19         List<Integer> array2 = new ArrayList<>();
20         for (int i = 0; i < size2; i++) {
21             array2.add(scanner.nextInt());
22         }
23
24         // 输入需要取出的元素对数
25         int k = scanner.nextInt();
26
27

```

```

27 // 存储所有可能的元素对的和
28 List<Integer> pairsSum = new ArrayList<>();
29 for (int value1 : array1) {
30     for (int value2 : array2) {
31         pairsSum.add(value1 + value2);
32     }
33 }
34
35 // 对和进行排序
36 Collections.sort(pairsSum);
37
38 // 取前k个元素进行求和
39 int minSum = 0;
40 for (int i = 0; i < k; i++) {
41     minSum += pairsSum.get(i);
42 }
43
44 System.out.println(minSum);
45 }
46 }
47
48

```

Python

```

1 # 从输入中获取数组array1, 使用map函数将输入的字符串转换为整数, 并使用列表切片[1:]去除第一个元素
2 array1 = list(map(int, input().split()[1:]))
3
4 # 从输入中获取数组array2, 使用map函数将输入的字符串转换为整数, 并使用列表切片[1:]去除第一个元素
5 array2 = list(map(int, input().split()[1:]))
6
7 # 从输入中获取k的值, 将其转换为整数
8 k = int(input())
9
10 # 存储所有可能的元素对的和
11 pairsSum = []
12 for value1 in array1:
13     for value2 in array2:
14         pairsSum.append(value1 + value2)
15
16

```

```

16 # 对和进行排序
17 pairsSum.sort()
18
19 # 取前k个元素进行求和
20 minSum = sum(pairsSum[:k])
21
22 # 输出最小和
23 print(minSum)
24
25

```

C语言

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  // 比较函数, 用于qsort
5  int compare(const void *a, const void *b) {
6      return (*(int*)a - *(int*)b);
7  }
8
9  int main() {
10     int size1, size2, k;
11     // 输入数组array1和array2的大小
12     scanf("%d", &size1);
13     int array1[size1];
14     for (int i = 0; i < size1; i++) {
15         scanf("%d", &array1[i]);
16     }
17
18     scanf("%d", &size2);
19     int array2[size2];
20     for (int i = 0; i < size2; i++) {
21         scanf("%d", &array2[i]);
22     }
23
24     // 输入k的值
25     scanf("%d", &k);
26
27     // 创建数组来存储所有可能的元素对的和
28

```

```

28     int pairsSum[size1 * size2];
29     int count = 0;
30     for (int i = 0; i < size1; i++) {
31         for (int j = 0; j < size2; j++) {
32             // 将array1和array2中的元素两两相加, 并将结果存储到pairsSum中
33             pairsSum[count++] = array1[i] + array2[j];
34         }
35     }
36
37     // 对pairsSum中的元素进行排序
38     qsort(pairsSum, count, sizeof(int), compare);
39
40     // 计算前k个元素的和
41     int minSum = 0;
42     for (int i = 0; i < k; i++) {
43         minSum += pairsSum[i];
44     }
45
46     // 输出最小和
47     printf("%d\n", minSum);
48
49     return 0;
50 }

```

完整用例

用例1

1	1 1
2	1 1
3	1

用例2

1	3 1 1 1
2	3 1 1 1
3	3

用例3

1	3	1	2	3
2	3	4	5	6
3	2			

用例4

1	3	3	2	1
2	3	6	5	4
3	3			

用例5

1	4	1	1	1	1
2	4	2	3	4	5
3	4				

用例6

1	3	1	2	3
2	3	4	5	6
3	9			

用例7

1	3	1	2	3
2	3	4	5	6
3	3			

用例8

1	3	1	2	3
2	3	4	5	6
3	1			

用例9

1	3	1	2	3
2	3	4	5	6
3				

用例10

1	3 1 2 3
2	3 4 5 6
3	7

文章目录

- 华为OD机考:统一考试 C卷 + D卷 + B卷 +A卷
 - 题目描述: 整数对最小和 (分值100)
 - 输入描述
 - 输出描述
 - 用例
 - C++
 - JavaScript
 - Java
 - Python
 - C语言
 - 完整用例
 - 用例1
 - 用例2
 - 用例3
 - 用例4
 - 用例5
 - 用例6
 - 用例7
 - 用例8
 - 用例9
 - 用例10

机考真题 华为OD



CSDN @算法大师