

【华为OD机考 统一考试机试C卷】求最多可以派出多少支团队（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

用数组代表每个人的能力 一个比赛活动要求参赛团队的最低能力值为N 每个团队可以由一人或者两人组成 且一个人只能参加一个团队 计算出最多可以派出多少只符合要求的队伍。

输入描述

第一行代表总人数，范围1-500000

第二行数组代表每个人的能力

- 数组大小，范围1-500000

- 元素取值，范围1-500000

第三行数值为团队要求的最低能力值，范围1-500000

输出描述

最多可以派出的团队数量

用例1

输入

1	5
2	3 1 5 7 9
3	8

输出

1	3
---	---

说明

3、5组成一队 1、7一队 9自己一队 输出3

用例2

输入

1	7
2	3 1 5 7 9 2 6
3	8

输出

1	4
---	---

说明

3、5组成一队， 1、7一队， 9自己一队， 2、6一队， 输出4

用例3

输入

```
1 | 3
2 | 1 1 9
3 | 8
```

输出

```
1 | 1
```

说明

9自己一队，输出1

C++

```
1 |
2 | #include <iostream>
3 | #include <vector>
4 | #include <algorithm>
5 | using namespace std;
6 |
7 | int main() {
8 |     int n; // 总人数
9 |     cin >> n;
10 |
11 |     vector<int> vec(n); // 每个人的能力
12 |     for (int i = 0; i < n; i++) {
13 |         cin >> vec[i];
14 |     }
15 |
16 |     sort(vec.begin(), vec.end()); // 对能力进行排序
17 |
18 |     int m; // 团队要求的最低能力值
19 |     cin >> m;
20 |
21 |     int left = 0; // 左指针
22 |     int right = n - 1; // 右指针
23 |     int res = 0; // 最多可以派出的团队数量
24 |
```

```

25
26 while (left < right) {
27     if (vec[right] >= m) { // 如果右指针指向的人的能力大于等于团队要求的最低能力值
28         res += 1; // 可以派出一个团队
29         right -= 1; // 右指针向左移动
30     } else if (vec[right] + vec[left] >= m) { // 如果右指针和左指针指向的两个人的能力之和大于等于团队要求的最低能力值
31         res += 1; // 可以派出一个团队
32         right -= 1; // 右指针向左移动
33         left += 1; // 左指针向右移动
34     } else { // 如果右指针和左指针指向的两个人的能力之和小于团队要求的最低能力值
35         left += 1; // 左指针向右移动
36     }
37 }
38
39 if (left == right && vec[left] >= m) { // 如果左指针和右指针指向同一个人，并且该人的能力大于等于团队要求的最低能力值
40     res += 1; // 可以派出一个团队
41 }
42
43 cout << res << endl; // 输出最多可以派出的团队数量
44
45 return 0;
}

```

java

```

1 import java.util.Arrays;
2 import java.util.Scanner;
3
4 public class Main {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7         int n = scanner.nextInt(); // 总人数
8         int[] vec = new int[n]; // 每个人的能力
9         for (int i = 0; i < n; i++) {
10             vec[i] = scanner.nextInt();
11         }
12         Arrays.sort(vec); // 对能力进行排序
13         int m = scanner.nextInt(); // 团队要求的最低能力值
14         int left = 0; // 左指针
15         int right = n - 1; // 右指针
16

```

```

16     int res = 0; // 最多可以派出的团队数量
17
18     while (left < right) {
19         if (vec[right] >= m) { // 如果右指针指向的人的能力大于等于团队要求的最低能力值
20             res += 1; // 可以派出一个团队
21             right -= 1; // 右指针向左移动
22         } else if (vec[right] + vec[left] >= m) { // 如果右指针和左指针指向的两个人的能力之和大于等于团队要求的最低能力值
23             res += 1; // 可以派出一个团队
24             right -= 1; // 右指针向左移动
25             left += 1; // 左指针向右移动
26         } else { // 如果右指针和左指针指向的两个人的能力之和小于团队要求的最低能力值
27             left += 1; // 左指针向右移动
28         }
29     }
30
31     if (left == right && vec[left] >= m) { // 如果左指针和右指针指向同一个人，并且该人的能力大于等于团队要求的最低能力值
32         res += 1; // 可以派出一个团队
33     }
34
35     System.out.println(res); // 输出最多可以派出的团队数量
36 }
37 }

```

javaScript

```

1
2 const readline = require('readline');
3
4 const rl = readline.createInterface({
5     input: process.stdin,
6     output: process.stdout
7 });
8
9 let totalPeople;
10 let abilities;
11 let minAbility;
12
13 rl.on('line', (line) => {
14     if (!totalPeople) {
15         totalPeople = parseInt(line);
16     }
17 }

```

```

16 } else if (!abilities) {
17   abilities = line.split(' ').map(Number);
18 } else if (!minAbility) {
19   minAbility = parseInt(line);
20
21   abilities.sort((a,b)=>a -b );
22
23   let left = 0;
24   let right = totalPeople - 1;
25   let res = 0;
26
27   while (left < right) {
28     if (abilities[right] >= minAbility) {
29       res += 1;
30       right -= 1;
31     } else if (abilities[right] + abilities[left] >= minAbility) {
32       res += 1;
33       right -= 1;
34       left += 1;
35     } else {
36       left += 1;
37     }
38   }
39
40   if (left === right && abilities[left] >= minAbility) {
41     res += 1;
42   }
43
44   console.log(res);
45 }
46 });

```

python

```

1
2 n = int(input()) # 总人数
3 vec = list(map(int, input().split())) # 每个人的能力
4 vec.sort() # 对能力进行排序
5 m = int(input()) # 团队要求的最低能力值
6 left = 0 # 左指针
7

```

```

7 | right = n - 1 # 右指针
8 | res = 0 # 最多可以派出的团队数量
9 |
10 | while left < right:
11 |     if vec[right] >= m: # 如果右指针指向的人的能力大于等于团队要求的最低能力值
12 |         res += 1 # 可以派出一个团队
13 |         right -= 1 # 右指针向左移动
14 |     elif vec[right] + vec[left] >= m: # 如果右指针和左指针指向的两个人的能力之和大于等于团队要求的最低能力值
15 |         res += 1 # 可以派出一个团队
16 |         right -= 1 # 右指针向左移动
17 |         left += 1 # 左指针向右移动
18 |     else: # 如果右指针和左指针指向的两个人的能力之和小于团队要求的最低能力值
19 |         left += 1 # 左指针向右移动
20 |
21 | if left == right and vec[left] >= m: # 如果左指针和右指针指向同一个人，并且该人的能力大于等于团队要求的最低能力值
22 |     res += 1 # 可以派出一个团队
23 |
24 | print(res) # 输出最多可以派出的团队数量

```

C语言

```

1 | #include <stdio.h>
2 | #include <stdlib.h>
3 |
4 | // 比较函数, 用于qsort排序
5 | int cmp(const void *a, const void *b) {
6 |     return (*(int *)a - *(int *)b);
7 | }
8 |
9 | int main() {
10 |     int n; // 总人数
11 |     scanf("%d", &n);
12 |
13 |     int *vec = (int *)malloc(n * sizeof(int)); // 动态分配数组存储每个人的能力
14 |     for (int i = 0; i < n; i++) {
15 |         scanf("%d", &vec[i]);
16 |     }
17 |
18 |     qsort(vec, n, sizeof(int), cmp); // 使用qsort对能力进行排序
19 |
20 |

```

```

20     int m; // 团队要求的最低能力值
21     scanf("%d", &m);
22
23     int left = 0; // 左指针
24     int right = n - 1; // 右指针
25     int res = 0; // 最多可以派出的团队数量
26
27     while (left < right) {
28         if (vec[right] >= m) { // 如果右指针指向的人的能力大于等于团队要求的最低能力值
29             res++; // 可以派出一个团队
30             right--; // 右指针向左移动
31         } else if (vec[right] + vec[left] >= m) { // 如果右指针和左指针指向的两个人的能力之和大于等于团队要求的最低能力值
32             res++; // 可以派出一个团队
33             right--; // 右指针向左移动
34             left++; // 左指针向右移动
35         } else { // 如果右指针和左指针指向的两个人的能力之和小于团队要求的最低能力值
36             left++; // 左指针向右移动
37         }
38     }
39
40     if (left == right && vec[left] >= m) { // 如果左指针和右指针指向同一个人，并且该人的能力大于等于团队要求的最低能力值
41         res++; // 可以派出一个团队
42     }
43
44     printf("%d\n", res); // 输出最多可以派出的团队数量
45
46     free(vec); // 释放动态分配的内存
47     return 0;
48 }

```

完整用例

用例1

1	5
2	3 1 5 7 9
3	8

用例2

1	7
2	3 1 5 7 9 2 6
3	8

用例3

1	3
2	1 1 9
3	8

用例4

1	1
2	10
3	10

用例5

1	4
2	1 2 3 4
3	5

用例6

1	10
2	1 2 3 4 5 6 7 8 9 10
3	5

用例7

1	8
2	1 2 3 4 5 6 7 8
3	10

用例8

1	12
2	1 2 3 4 5 6 7 8 9 10 11 12
3	

用例9

1	6
2	10 10 10 10 10 10 10
3	5

用例10

1	15
2	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
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
 - 输入描述
 - 输出描述
 - 用例1
 - 用例2
 - 用例3
 - C++
 - java
 - javaScript
 - 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