

【华为OD机考 统一考试机试C卷】开源项目热度榜单 (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

某个开源社区希望将最近热度比较高的开源项目出一个榜单，推荐给社区里面的开发者。对于每个开源项目，开发者可以进行关注(watch)、收藏(star)、fork、提issue、提交合并请求(MR)等。

数据库里面统计了每个开源项目关注、收藏、fork、issue、MR的数量，开源项目的热度根据这5个维度的加权求和进行排序。

$$1 \quad H = (W_{\text{watch}} * \#_{\text{watch}}) + (W_{\text{star}} * \#_{\text{star}}) + (W_{\text{fork}} * \#_{\text{fork}}) + (W_{\text{issue}} * \#_{\text{issue}}) + (W_{\text{mr}} * \#_{\text{mr}})$$

H表示热度值

W_{watch} 、 W_{star} 、 W_{fork} 、 W_{issue} 、 W_{mr} 分别表示5个统计维度的权重。

$\#_{\text{watch}}$ 、 $\#_{\text{star}}$ 、 $\#_{\text{fork}}$ 、 $\#_{\text{issue}}$ 、 $\#_{\text{mr}}$ 分别表示5个统计维度的统计值。

榜单按照热度值降序排序，对于热度值相等的，按照项目名字转换为全小写字母后的字典序排序 ('a','b','c',...,'x','y','z')。

输入描述

第一行输入为N，表示开源项目的个数， $0 < N < 100$ 。

第二行输入为权重值列表，一共 5 个整型值，分别对应关注、收藏、fork、issue、MR的权重，权重取值 $0 < W \leq 50$ 。

第三行开始接下来的 N 行为开源项目的统计维度，每一行的格式为：

```
1 | name nr_watch nr_start nr_fork nr_issue nr_mr
```

其中 name 为开源项目的名字，由英文字母组成，长度 ≤ 50 ，其余 5 个整型值分别为该开源项目关注、收藏、fork、issue、MR的数量，数量取值 $0 < nr \leq 1000$ 。

输出描述

按照热度降序，输出开源项目的名字，对于热度值相等的，按照项目名字转换为全小写后的字典序排序（'a' > 'b' > 'c' > ... > 'x' > 'y' > 'z'）。

用例1

输入

```
1 | 4
2 | 8 6 2 8 6
3 | camila 66 70 46 158 80
4 | victoria 94 76 86 189 211
5 | anthony 29 17 83 21 48
6 | emily 53 97 1 19 218
```

输出

```
1 | victoria
2 | camila
3 | emily
4 | anthony
```

说明

排序热度值计算：
camila: $66*8 + 70*6 + 46*2 + 158*8 + 80*6 = 2784$
victoria: $94*8 + 76*6 + 86*2 + 189*8 + 211*6 = 4158$

anthony: $29*8 + 17*6 + 83*2 + 21*8 + 48*6 = 956$
emily: $53*8 + 97*6 + 1*2 + 19*8 + 218*6 = 2468$

用例2

输入

```
1 5
2 5 6 6 1 2
3 camila 13 88 46 26 169
4 grace 64 38 87 23 103
5 lucas 91 79 98 154 79
6 leo 29 27 36 43 178
7 ava 29 27 36 43 178
```

输出

```
1 lucas
2 grace
3 camila
4 ava
5 leo
```

说明

排序热度值计算：

camila: $13*5 + 88*6 + 46*6 + 26*1 + 169*2 = 1233$

grace: $64*5 + 38*6 + 87*6 + 23*1 + 103*2 = 1299$

lucas: $91*5 + 79*6 + 98*6 + 154*1 + 79*2 = 1829$

leo: $29*5 + 27*6 + 36*6 + 43*1 + 178*2 = 922$

ava: $29*5 + 27*6 + 36*6 + 43*1 + 178*2 = 922$

解题思路

2. 读取其名称和评分。然后，我们使用权重和评分来计算项目的"热度"。这是通过将每个评分与其对应的权重相乘，然后将所有的乘积相加来完成的。

3. 排序首先是根据热度进行的，热度高的项目排在前面。如果两个项目的热度相同，那么我们就根据项目名称进行排序，名称按字母顺序排列。

C++

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  #include <string>
5
6  using namespace std;
7
8  // 定义一个结构体来存储项目的名称和热度
9  struct Project {
10     string name;
11     int hotness;
12 };
13
14 // 自定义比较函数，用于排序
15 bool compare(Project a, Project b) {
16     if (a.hotness != b.hotness) {
17         return a.hotness > b.hotness; // 热度高的项目排在前面
18     } else {
19         return a.name < b.name; // 名称相同的项目按字母顺序排列
20     }
21 }
22
23 int main() {
24     int n;
25     cin >> n;
26
27     vector<int> weights(5);
28     for (int i = 0; i < 5; i++) {
29         cin >> weights[i]; // 读取每个权重并存储到数组中
30     }
31
32     vector<Project> projects(n);
33     for (int i = 0; i < n; i++) {
34         cin >> projects[i].name; // 存储项目名称
35     }
```

```

36
37     int hot = 0;
38     for (int j = 0; j < 5; j++) {
39         int score;
40         cin >> score;
41         hot += score * weights[j]; // 计算热度
42     }
43     projects[i].hotness = hot;
44 }
45
46 sort(projects.begin(), projects.end(), compare); // 使用自定义比较函数进行排序
47
48 for (Project project : projects) {
49     cout << project.name << endl; // 打印项目名称
50 }
51
52 return 0;
53 }

```

Java

```

1 import java.util.Arrays;
2 import java.util.Comparator;
3 import java.util.Scanner;
4
5 public class Main {
6     public static void main(String[] args) {
7         // 创建Scanner对象用于获取用户输入
8         Scanner sc = new Scanner(System.in);
9
10        // 读取项目数量n
11        int n = sc.nextInt();
12
13        // 创建并填充权重数组weights
14        int[] weights = new int[5];
15        for (int i = 0; i < 5; i++) {
16            weights[i] = sc.nextInt(); // 读取每个权重并存储到数组中
17        }
18
19        // 创建一个二维数组projects来存储项目的名称和热度值
20
21    }
22 }

```

```

20 // 其中projects[i][0]存储项目名称, projects[i][1]存储项目热度
21 String[][] projects = new String[n][2];
22
23 // 读取项目信息并计算每个项目的热度
24 for (int i = 0; i < n; i++) {
25     projects[i][0] = sc.next(); // 存储项目名称
26
27     int hot = 0; // 初始化项目热度为0
28     // 读取项目的5个评分并计算热度
29     for (int j = 0; j < 5; j++) {
30         hot += sc.nextInt() * weights[j]; // 计算热度
31     }
32     projects[i][1] = String.valueOf(hot); // 将热度值转换为字符串并存储
33 }
34
35 // 使用自定义比较器对项目数组进行排序
36 Arrays.sort(projects, new Comparator<String[]>() {
37     @Override
38     public int compare(String[] a, String[] b) {
39         // 解析字符串热度为整数
40         int hotA = Integer.parseInt(a[1]);
41         int hotB = Integer.parseInt(b[1]);
42         // 首先根据热度值降序排序
43         if (hotA != hotB) {
44             return hotB - hotA; // 热度高的项目排在前面
45         } else {
46             // 如果热度相同, 则根据项目名称字母顺序升序排序
47             return a[0].toLowerCase().compareTo(b[0].toLowerCase()); // 名称相同的项目按字母顺序排列
48         }
49     }
50 });
51
52 // 遍历排序后的项目数组并打印项目名称
53 for (String[] project : projects) {
54     System.out.println(project[0]); // 打印项目名称
55 }
56
57 // 关闭Scanner对象, 释放资源
58 sc.close();
59
60

```

```
}  
}
```

JavaScript

```
1  const readline = require('readline');  
2  
3  const rl = readline.createInterface({  
4    input: process.stdin,  
5    output: process.stdout  
6  });  
7  
8  // 创建一个数组用于存储用户的输入  
9  let input = [];  
10 // 当接收到一行输入时, 将其去除首尾空格后添加到input数组中  
11 rl.on('line', (line) => {  
12   input.push(line.trim());  
13 }).on('close', () => { // 当输入结束时, 执行以下代码  
14   // 读取项目数量n  
15   const n = parseInt(input[0]);  
16   // 读取权重数组, 将其转换为数字数组  
17   const weights = input[1].split(' ').map(Number);  
18  
19   // 创建一个数组用于存储项目信息  
20   let projects = [];  
21   // 读取每个项目的信息  
22   for (let i = 2; i < 2 + n; i++) {  
23     // 将项目信息分割为名称和评分数组  
24     const project = input[i].split(' ');  
25     const name = project[0];  
26     const scores = project.slice(1).map(Number);  
27  
28     // 计算项目的热度  
29     let hotness = 0;  
30     for (let j = 0; j < 5; j++) {  
31       hotness += scores[j] * weights[j];  
32     }  
33  
34     // 将项目的名称和热度添加到projects数组中  
35     projects.push({ name, hotness });  
36   }
```

```

36     }
37
38     // 对项目数组进行排序, 首先根据热度降序排序, 如果热度相同则根据名称升序排序
39     projects.sort((a, b) => {
40         if (a.hotness !== b.hotness) {
41             return b.hotness - a.hotness;
42         } else {
43             return a.name.localeCompare(b.name);
44         }
45     });
46
47     // 遍历排序后的项目数组并打印项目名称
48     for (let project of projects) {
49         console.log(project.name);
50     }
51 });

```

Python

```

1  # 读取第一行输入, 转换为整数, 表示项目的数量
2  n = int(input())
3  # 读取第二行输入, 分割字符串并将每个元素转换为整数, 得到权重列表
4  weights = list(map(int, input().split()))
5
6  # 初始化一个空列表, 用于存储项目的名称和计算出的热度
7  projects = []
8  # 循环n次, 对应于n个项目
9  for _ in range(n):
10     # 读取每个项目的输入, 分割字符串得到项目名称和评分
11     project = input().split()
12     name = project[0] # 第一个元素是项目名称
13     scores = list(map(int, project[1:])) # 剩余元素是项目的评分, 转换为整数列表
14
15     # 使用列表推导式和zip函数计算项目的热度
16     # zip函数将评分和权重配对, 列表推导式计算乘积并求和
17     hotness = sum(score * weight for score, weight in zip(scores, weights))
18
19     # 将项目名称和热度作为元组添加到projects列表中
20     projects.append((name, hotness))
21
22

```



```

42 # 对projects列表进行排序
43 # 使用lambda函数作为排序的关键字, 首先根据热度降序(-x[1])排序,
44 # 如果热度相同, 则根据项目名称升序(x[0])排序
45 projects.sort(key=lambda x: (-x[1], x[0]))
46
47 # 遍历排序后的项目列表
48 for project in projects:
49     print(project[0]) # 打印每个项目的名称

```

C语言

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  // 定义一个结构体来存储项目的名称和热度
6  typedef struct {
7      char name[51];
8      int hotness;
9  } Project;
10
11 // 自定义比较函数, 用于排序
12 int compare(const void *a, const void *b) {
13     Project *projectA = (Project *)a;
14     Project *projectB = (Project *)b;
15     if (projectA->hotness != projectB->hotness) {
16         return projectB->hotness - projectA->hotness; // 热度高的项目排在前面
17     } else {
18         return strcmp(projectA->name, projectB->name); // 名称相同的项目按字母顺序排列
19     }
20 }
21
22 int main() {
23     int n;
24     scanf("%d", &n);
25
26     int weights[5];
27     for (int i = 0; i < 5; i++) {
28         scanf("%d", &weights[i]); // 读取每个权重
29     }
30

```

```

30
31     Project projects[n];
32     for (int i = 0; i < n; i++) {
33         scanf("%s", projects[i].name); // 存储项目名称
34
35         int hot = 0;
36         for (int j = 0; j < 5; j++) {
37             int score;
38             scanf("%d", &score);
39             hot += score * weights[j]; // 计算热度
40         }
41         projects[i].hotness = hot;
42     }
43
44     qsort(projects, n, sizeof(Project), compare); // 使用自定义比较函数进行排序
45
46     for (int i = 0; i < n; i++) {
47         printf("%s\n", projects[i].name); // 打印项目名称
48     }
49
50     return 0;
51 }

```

完整用例

用例1

```

1 3
2 10 20 30 40 50
3 Alpha 10 20 30 40 50
4 Beta 20 30 10 50 40
5 Gamma 20 10 40 30 50

```

用例2

```

1 4
2 5 5 5 5 5
3 ProjectX 100 200 300 400 500
4 ProjectY 500 400 300 200 100
5

```

```
~ |
6 | ProjectZ 250 250 250 250 250
   Delta 300 400 200 100 500
```

用例3

```
1 | 6
2 | 1 1 1 1 1
3 | RepoA 1 2 3 4 5
4 | RepoB 5 4 3 2 1
5 | RepoC 2 3 4 5 6
6 | RepoD 6 5 4 3 2
7 | RepoE 3 3 3 3 3
8 | RepoF 4 4 4 4 4
```

用例4

```
1 | 5
2 | 15 25 35 20 10
3 | LibOne 10 10 10 10 10
4 | LibTwo 20 20 20 20 20
5 | LibThree 5 5 5 5 5
6 | LibFour 15 15 15 15 15
7 | LibFive 25 25 25 25 25
```

用例5

```
1 | 3
2 | 50 40 30 20 10
3 | ModuleA 50 40 30 20 10
4 | ModuleB 40 30 20 10 50
5 | ModuleC 10 20 30 40 50
```

用例6

```
1 | 4
2 | 7 8 9 10 11
3 | ToolX 9 8 7 6 5
4 | ToolY 5 6 7 8 9
5 |
~ |
```

```
6 | ToolZ 8 9 10 11 12
   | ToolA 12 11 10 9 8
```

用例7

```
1 | 2
2 | 50 1 50 1 50
3 | CodeBase 100 100 100 100 100
4 | CodePeak 100 100 100 100 100
```

用例8

```
1 | 7
2 | 10 20 30 40 50
3 | Query 1 2 3 4 5
4 | Search 2 3 4 5 1
5 | Find 3 4 5 1 2
6 | Locate 4 5 1 2 3
7 | Discover 5 1 2 3 4
8 | Explore 1 3 5 2 4
9 | Investigate 2 4 1 3 5
```

用例9

```
1 | 5
2 | 8 6 4 2 1
3 | FrameworkA 88 77 66 55 44
4 | FrameworkB 77 88 99 11 22
5 | FrameworkC 66 55 44 33 11
6 | FrameworkD 55 44 33 22 11
7 | FrameworkE 44 33 22 11 99
```

用例10

```
1 | 6
2 | 45 35 25 15 5
3 | PackageOne 10 9 8 7 6
4 | PackageTwo 9 8 7 6 5
5 | PackageThree 8 7 6 5 4
```

| | | |
|---|--|-----------------------|
| 6 | | PackageFour 7 6 5 4 3 |
| 7 | | PackageFive 6 5 4 3 2 |
| 8 | | PackageSix 5 4 3 2 1 |

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](#)

[解题思路](#)

[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