

【华为OD机考 统一考试机试C卷】 找朋友 (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

在学校中，N个小朋友站成一队，第i个小朋友的身高为height[i]，

第i个小朋友可以看到的第一个比自己身高更高的小朋友j，那么j是i的好朋友(要求j > i)。

请重新生成一个列表，对应位置的输出是每个小朋友的好朋友位置，如果没有看到好朋友，请在该位置用0代替。

小朋友人数范围是 [0, 40000]。

输入描述

第一行输入N，N表示有N个小朋友

第二行输入N个小朋友的身高height[i]，都是整数

输出描述

输出N个小朋友的好朋友的位置

用例1

输入

```
1 | 2
2 | 100 95
```

输出

```
1 | 0 0
```

说明

第一个小朋友身高100，站在队尾位置，向队首看，没有比他身高高的小朋友，所以输出第一个值为0。
第二个小朋友站在队首，前面也没有比他身高高的小朋友，所以输出第二个值为0。

用例2

输入

```
1 | 8
2 | 123 124 125 121 119 122 126 123
```

输出

```
1 | 1 2 6 5 5 6 0 0
```

说明

123的好朋友是1位置上的124
124的好朋友是2位置上的125
125的好朋友是6位置上的126
以此类推

C++

```

1  #include <iostream>
2  #include <vector>
3  #include <stack>
4  #include <sstream>
5
6  using namespace std;
7
8  int main() {
9      int n;
10     cin >> n;
11
12     vector<int> heights(n);
13     for (int i = 0; i < n; i++) {
14         cin >> heights[i];
15     }
16
17     vector<int> friendIndexes(n);
18     stack<int> stack;
19     stack.push(0);
20     for (int i = 1; i < n; i++) {
21         while (!stack.empty() && heights[i] > heights[stack.top()]) {
22             friendIndexes[stack.top()] = i;
23             stack.pop();
24         }
25         stack.push(i);
26     }
27
28     stringstream result;
29     for (int i = 0; i < n; i++) {
30         result << friendIndexes[i] << " ";
31     }
32     cout << result.str().substr(0, result.str().size() - 1) << endl;
33
34     return 0;
35 }

```

javaScript

```

1  const readline = require('readline');
2  const rl = readline.createInterface({

```

```

3   input: process.stdin,
4   output: process.stdout
5 });
6
7 let n = 0;
8 let height = [];
9
10 rl.on('line', (input) => {
11   if (!n) {
12     n = parseInt(input.trim());
13   } else {
14     height = input.trim().split(' ').map(Number);
15
16     let friendIndexes = new Array(n).fill(0);
17     let stack = [0];
18     for (let i = 1; i < n; i++) {
19       while (stack.length && height[i] > height[stack[stack.length - 1]]) {
20         friendIndexes[stack.pop()] = i;
21       }
22       stack.push(i);
23     }
24
25     let result = "";
26     for (let i = 0; i < n; i++) {
27       result += friendIndexes[i] + " ";
28     }
29     console.log(result.trim());
30   }
31 });

```

java

```

1 import java.util.Scanner;
2 import java.util.Stack;
3
4 public class Main {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7
8         // 输入小朋友的数量
9

```

```

9      int n = Integer.parseInt(scanner.nextLine());
10
11      // 输入小朋友的身高
12      int[] heights = new int[n];
13      String[] input = scanner.nextLine().split(" ");
14      for (int i = 0; i < n; i++) {
15          heights[i] = Integer.parseInt(input[i]);
16      }
17
18      // 用于存储每个小朋友的好朋友位置
19      int[] friendIndexes = new int[n];
20
21      // 使用栈来记录每个小朋友的位置
22      Stack<Integer> stack = new Stack<>();
23      stack.push(0);
24      for (int i = 1; i < n; i++) {
25          while (!stack.isEmpty() && heights[i] > heights[stack.peek()]) {
26              // 如果当前小朋友的身高大于栈顶小朋友的身高，则栈顶小朋友的好朋友位置为当前小朋友的位置
27              friendIndexes[stack.pop()] = i;
28          }
29          stack.push(i);
30      }
31
32      // 输出每个小朋友的好朋友位置
33      StringBuilder result = new StringBuilder();
34      for (int i = 0; i < n; i++) {
35          result.append(friendIndexes[i]).append(" ");
36      }
37      System.out.println(result.toString().trim());
38  }
39 }

```

python

```

1  import sys
2
3  # 输入小朋友的数量
4  n = int(input())
5
6  # 输入小朋友的身高
7

```

```

7 | heights = list(map(int, input().split()))
8 |
9 | # 用于存储每个小朋友的好朋友位置
10 | friendIndexes = [0] * n
11 |
12 | # 使用栈来记录每个小朋友的位置
13 | stack = [0]
14 | for i in range(1, n):
15 |     while stack and heights[i] > heights[stack[-1]]:
16 |         # 如果当前小朋友的身高大于栈顶小朋友的身高，则栈顶小朋友的好朋友位置为当前小朋友的位置
17 |         friendIndexes[stack.pop()] = i
18 |     stack.append(i)
19 |
20 | # 输出每个小朋友的好朋友位置
21 | result = ""
22 | for i in range(n):
23 |     result += str(friendIndexes[i]) + " "
24 | print(result.strip())

```

C语言

```

1 | #include <stdio.h>
2 |
3 | #define MAX_N 40000
4 |
5 | int main() {
6 |     int n;
7 |     scanf("%d", &n); // 读取小朋友的数量
8 |
9 |     int heights[MAX_N]; // 存储每个小朋友的身高
10 |    for (int i = 0; i < n; i++) {
11 |        scanf("%d", &heights[i]); // 读取每个小朋友的身高
12 |    }
13 |
14 |    int friendIndexes[MAX_N] = {0}; // 存储每个小朋友的好朋友的位置，初始化为0
15 |    int stack[MAX_N]; // 存储小朋友的索引，用作栈
16 |    int top = 0; // 栈顶指针
17 |    stack[top] = 0; // 将第一个小朋友的索引压入栈
18 |
19 |    for (int i = 1; i < n; i++) {
20 |

```

```

20         // 当栈不为空且当前小朋友的身高大于栈顶小朋友的身高时
21         while (top >= 0 && heights[i] > heights[stack[top]]) {
22             // 更新栈顶小朋友的好朋友的位置为当前小朋友的位置
23             friendIndexes[stack[top]] = i;
24             // 弹出栈顶元素
25             top--;
26         }
27         // 将当前小朋友的索引压入栈
28         stack[++top] = i;
29     }
30
31     // 输出每个小朋友的好朋友的位置
32     for (int i = 0; i < n; i++) {
33         printf("%d ", friendIndexes[i]);
34     }
35
36     return 0;
37 }

```

完整用例

用例1

```

1 | 2
2 | 100 95

```

用例2

```

1 | 5
2 | 120 125 130 115 110

```

用例3

```

1 | 3
2 | 105 100 110

```

用例4

1	4
2	115 120 125 110

用例5

1	6
2	130 125 120 135 140 115

用例6

1	8
2	123 124 125 121 119 122 126 123

用例7

1	10
2	130 135 140 125 120 115 110 145 150 105

用例8

1	15
2	120 125 130 135 140 115 110 105 100 145 150 155 160 165 170

用例9

1	6
2	100 95 105 90 110 85

用例10

1	12
2	120 125 130 135 140 115 110 105 100 145 150 155

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

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