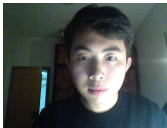


校招全国统一模拟笔试（2020年2月场）- C++方向测评结果

考生信息



邓世杰

投递编号：573985699 | 学校：湖南大学 | 邮箱：dengshijie@hnu.edu.cn | 手机号码：13237612997 |

学历：硕士 | 职位：C++ | 参考区域：江西省南昌市（49.211.155.173） |

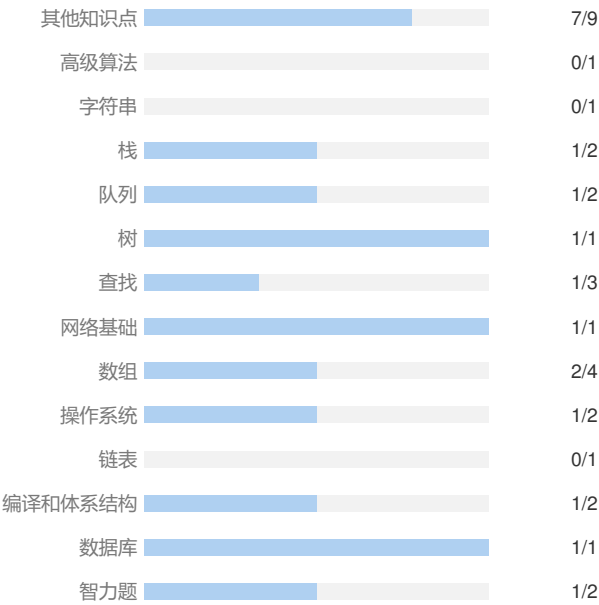
做题用时：01:57:10(2020-02-27 19:00:05开始答题，20:57:21交卷) | 作答设备：PC

考生成绩



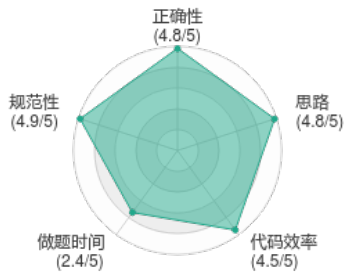
题型	得分	正确题数	排名	用时	是否阅卷
单选	26.0	13	191	00:36:07	--
编程	58.0	2	13	01:17:29	--

知识点技能图谱



知识点	得分	正确题数
其他知识点	68.0	7
高级算法	0.0	0
字符串	0.0	0
栈	2.0	1
队列	2.0	1
树	2.0	1
查找	2.0	1
网络基础	2.0	1
数组	4.0	2
操作系统	2.0	1
链表	0.0	0
编译和体系结构	2.0	1
数据库	2.0	1
智力题	2.0	1

编码能力



题号	正确性	提交次数	做题用时	使用语言	运行时间	占用内存	编程思路	代码规范	成绩排名
编程题1	100%	1	00:22:18	C++	45ms	1256K			1%
编程题2	100%	3	00:18:23	C++	178ms	1272K			1%
编程题3	90%	3	00:36:48	C++	57ms	3176K			7%

1 [平均分1.3分 | 211人正确/336人做题 | 用时：2分] 得分：2.0 / 2.0

有 ABCDEF 六个城市，每一个城市都和其他所有城市直接相连，问从 A — B 有多少种连接方式，路径不允许在两个城市之间往返。

- A 78
B 65
C 43
D 其他选项都不对

他的回答：B (正确)

正确答案：B

2 [平均分1.8分 | 311人正确/340人做题 | 用时：2分] 得分：2.0 / 2.0

甲有若干本书，乙借走了一半加 3 本，剩下的书，丙借走了 1/3 加 2 本，再剩下的书，丁借走了 1/4 加 1 本，最后甲还有 2 本书。问甲原来有多少本书？

- A 24
B 33
C 13
D 28

他的回答：A (正确)

正确答案：A

3 [平均分0.8分 | 134人正确/335人做题 | 用时：15分] 得分：0.0 / 2.0

在如下8*6的矩阵中，请计算从A移动到B一共有多少种走法，要求每次只能向上或向右移动一格，并且不能通过P ()

							B
				P			
A							

- A 702
B 626
C 456
D 680
E 568
F 492

他的回答：A (错误)

正确答案：F

4 [平均分1.6分 | 276人正确/341人做题 | 用时：<1分] 得分：0.0 / 2.0

对有序数组{2、11、15、19、30、32、61、72、88、90、96}进行二分查找，则成功找到15需比较（ ）次

- A 3
- B 4
- C 2
- D 5

他的回答：A (错误)

正确答案：C

5 [平均分1.4分 | 250人正确/345人做题 | 用时：<1分] 得分：2.0 / 2.0

众里寻他千百度，蓦然回首，那人却在灯火阑珊处。——辛弃疾《青玉案》描述的是()

- A 贪心
- B 回溯
- C 穷举
- D 分治
- E 递归

他的回答：B (正确)

正确答案：B

参考答案：

回溯算法是一种试探法，基本思路是：从一条路往前走，能进则进，不能进则退回来，换一条路再试。这正符合辛弃疾《青玉案》的笔意。

6 [平均分0.5分 | 84人正确/336人做题 | 用时：<1分] 得分：0.0 / 2.0

使用KMP算法在文本串S中找模式串P是一种常见的方法。假设S=P={xyxyxyxyx}，亦即将S对自己进行匹配，匹配过程中正确的next数组是_____。

- A 0,1,1,2,2,1,2,2,3
- B 0,1,2,2,3,1,2,2,3
- C 0,1,1,2,3,1,2,2,3
- D 0,1,1,2,3,1,1,2,3
- E 0,1,2,2,3,1,1,2,3
- F 0,1,2,2,2,1,1,2,3

他的回答：B (错误)

正确答案：C

7 [平均分1.4分 | 237人正确/344人做题 | 用时：<1分] 得分：2.0 / 2.0

图的广度优先搜索算法需使用的辅助数据结构为（ ）

- A 三元组
- B 队列
- C 二叉树
- D 栈

他的回答：B (正确)

正确答案：B

8 [平均分1.7分 | 283人正确/343人做题 | 用时：2分] 得分：2.0 / 2.0

A为整数数组，N为A的数组长度，请问执行以下代码，最坏情况下的时间复杂度为_____。

```
void fun(int A[], int n) {
```

```
for (int i = n - 1; i >= 1; i--) {
    for (int j = 0; j < i; j++) {
        if (A[j] > A[j+1]) {
            int tmp = A[j + 1];
            A[j + 1] = A[j];
            A[j] = tmp;
        }
    }
}
```

- A $O(N)$
- B $O(N^2)$
- C $O(N\log(N))$
- D $O(\log(N))$
- E $O(N^3)$
- F 无法确定

他的回答： B (正确)

正确答案： B

9 [平均分0.9分 | 149人正确/338人做题 | 用时：<1分] 得分：0.0 / 2.0

假设某段通信电文仅由 6 个字母 ABCDEF 组成，字母在电文中出现的频率分别为2，3，7，15，4，6。根据这些频率作为权值构造哈夫曼编码，最终构造出的哈夫曼树带权路径长度与字母 B 的哈夫曼编码分别为_____。(这里假定左节点的值小于右节点的值)

- A 86，1011
- B 70，1000
- C 86，0001
- D 70，0010
- E 92，1000
- F 92，0100

他的回答： B (错误)

正确答案： A

参考答案：
注解:构造结果

10 [平均分1.7分 | 290人正确/346人做题 | 用时：<1分] 得分：0.0 / 2.0

下面（ ）数据结构常用于函数调用。

- A 队列
- B 栈
- C 链表
- D 数组

他的回答： D (错误)

正确答案： B

11 [平均分1.5分 | 263人正确/340人做题 | 用时：<1分] 得分：2.0 / 2.0

关于cookie以下描述中不正确的是_____。

- A cookie附带于http请求中
- B cookie有大小限制
- C 用户可以主动禁止cookie
- D https协议下cookie是明文传递的

他的回答： D (正确)

正确答案：D

12 [平均分1.0分 | 177人正确/342人做题 | 用时：<1分 | 得分：2.0 / 2.0

程序的完整编译过程分为是：预处理，编译，汇编等，如下关于编译阶段的编译优化的说法中不正确的是（ ）。

- A 死代码删除指的是编译过程直接抛弃掉被注释的代码；
- B 函数内联可以避免函数调用中压栈和退栈的开销
- C for循环的循环控制变量通常很适合调度到寄存器访问
- D 强度削弱是指执行时间较短的指令等价的替代执行时间较长的指令

他的回答：A (正确)

正确答案：A

13 [平均分1.6分 | 276人正确/346人做题 | 用时：<1分 | 得分：0.0 / 2.0

以下那一个不是进程的基本状态 ()

- A 阻塞态
- B 执行态
- C 就绪态
- D 完成态

他的回答：C (错误)

正确答案：D

14 [平均分1.1分 | 188人正确/339人做题 | 用时：2分 | 得分：2.0 / 2.0

在5个页框上使用LRU页面替换算法，当页框初始为空时，引用序列为0、1、7、8、6、2、3、7、2、9、8、1、0、2，系统将发生（ ）次缺页

- A 13
- B 12
- C 11
- D 8

他的回答：C (正确)

正确答案：C

参考答案：

缺页为：0、1、7、8、6、2、3、9、8、1、0，共11次

15 [平均分1.5分 | 267人正确/347人做题 | 用时：<1分 | 得分：2.0 / 2.0

牛客网的数据库有一个paper表，现在要查询试卷名字包含"人人网"的所有数据，则sql语句应该为？

- A SELECT * FROM paper WHERE paper_name LIKE '人人网';
- B SELECT * FROM paper WHERE paper_name LIKE '%人人网';
- C SELECT * FROM paper WHERE paper_name LIKE '人人网%';
- D SELECT * FROM paper WHERE paper_name LIKE '%人人网%';

他的回答：D (正确)

正确答案：D

16 [平均分1.0分 | 166人正确/347人做题 | 用时：<1分 | 得分：0.0 / 2.0

bash环境中挂起当前进程的方式是（ ）

- A ctrl+x
- B ctrl+z
- C ctrl+c
- D ctrl+v

他的回答：D (错误)

正确答案：B

17 [平均分1.7分 | 298人正确/344人做题 | 用时：<1分 | 得分：2.0 / 2.0

考虑函数原型void test (int a , int b=7 , char z='*') , 下面的函数调用中 , 属于不合法调用的是 ()

A test (5)

B test (5,8)

C test (6 , '#')

D test (0,0 , '#')

他的回答：C (正确)

正确答案：C

18 [平均分1.9分 | 336人正确/346人做题 | 用时：<1分 | 得分：2.0 / 2.0

用于包含C++标准库iostream的预处理指令是 ()

A #define<iostream>

B #include<iostream>

C #define'iostream'

D #include'iostream'

他的回答：B (正确)

正确答案：B

19 [平均分1.9分 | 328人正确/344人做题 | 用时：<1分 | 得分：2.0 / 2.0

已知字符'A'的ASCII码值是65 , 且char c1='A' , c2='D' ; 则执行语句printf ("%d , %d\n" , c1 , c2-2) ; 后的输出结果是 ()

A A , D

B A , B

C 65,68

D 65,66

他的回答：D (正确)

正确答案：D

20 [平均分1.3分 | 225人正确/342人做题 | 用时：<1分 | 得分：2.0 / 2.0

以下能对二维数组a进行正确初始化的语句是 ()

A int a[2][]={{1,0,1},{5,2,3}};

B int a[][3]={{1,2,3},{4,5,6}};

C int a[2][4]={1,2,3},{4,5},{6}};

D int a[][3]={{1,0,1},{1,1}};

他的回答：B (正确)

正确答案：B

21 [平均分9.8分 | 77人正确/267人做题 | 提交：1次 | 得分：20.0 / 20.0

标题：排序子序列 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

【排序子序列】牛牛定义排序子序列为一个数组中一段连续的子序列,并且这段子序列是非递增或者非递减排序的。牛牛有一个长度为n的整数数组A,他现在有一个任务是数组A分为若干段排序子序列,牛牛想知道他最少可以把这个数组分为几段排序子序列。

如样例所示,牛牛可以把数组A划分为[1,2,3]和[2,2,1]两个排序子序列,至少需要划分为2个排序子序列,所以输出2

输入描述：

输入的第一行为一个正整数 $n(1 \leq n \leq 10^5)$

第二行包括 n 个整数 $A_i(1 \leq A_i \leq 10^9)$,表示数组 A 的每个数字。

输出描述：

输出一个整数表示牛牛可以将 A 最少划分为多少段排序子序列

示例1：

输入

6
1 2 3 2 2 1

输出

2

代码片段

功能实现			代码提交统计			代码执行统计		
	TA的	平均		TA的	平均	答案正确：1		
总通过率	100%	48%	使用语言	C++				
基本测试用例通过率	6/6 (100%)	54%	做题用时	00:22:18	00:31:06			
边缘测试用例通过率	4/4 (100%)	40%	提交次数	1	6			
代码效率					代码规范及可读性			
	TA的	参考				代码规范得分 5.0		
运行时间	45ms	1s						
占用内存	1256K	32768K						

他的代码：

做题用时: 22 分钟 语言：C++ 运行时间：45ms 占用内存：1256K 程序状态：答案正确

```
#include<iostream>
using namespace std;

int HowMany(int *a, int start, int n)
{
    if (start == n)
        return 0;
    if (start == n - 1)
        return 1;
    int index = start + 1;
    while (a[index] == a[start])
        index++;
    if (a[index] > a[index - 1]) // up
    {
        while (index < n && a[index] >= a[index - 1])
            index++;
    }
    else
    {
        while (index < n && a[index] <= a[index - 1])
            index++;
    }
    return 1 + HowMany(a, index, n);
}
```

```

}

int main()
{
    int num;
    cin >> num;
    int *a = new int[num];
    for (int i = 0; i < num; i++)
        cin >> a[i];
    cout << HowMany(a, 0, num) << endl;
    return 0;
}

```



[点此](#)或手机扫描二维码查看代码编写过程

22 [平均分10.1分 | 65人正确/218人做题 | 提交: 3 次] 得分 : 20.0 / 20.0

标题 : 组队竞赛 | 时间限制 : 1秒 | 内存限制 : 32768K | 语言限制 : 不限

【组队竞赛】牛牛举办了一次编程比赛,参加比赛的有 $3 \times n$ 个选手,每个选手都有一个水平值 a_i .现在要将这些选手进行组队,一共组成 n 个队伍,即每个队伍3人.牛牛发现队伍的水平值等于该队伍队员中第二高水平值。

例如:

一个队伍三个队员的水平值分别是3,3,3.那么队伍的水平值是3

一个队伍三个队员的水平值分别是3,2,3.那么队伍的水平值是3

一个队伍三个队员的水平值分别是1,5,2.那么队伍的水平值是2

为了让比赛更有看点,牛牛想安排队伍使所有队伍的水平值总和最大。

样例所示:

如果牛牛把6个队员划分到两个队伍

如果方案为:

team1:{1,2,5}, team2:{5,5,8}, 这时候水平值总和为7.

而如果方案为:

team1:{2,5,8}, team2:{1,5,5}, 这时候水平值总和为10.

没有比总和为10更大的方案,所以输出10.

输入描述 :

输入的第一行为一个正整数 $n(1 \leq n \leq 10^5)$

第二行包括 $3 \times n$ 个整数 $a_i(1 \leq a_i \leq 10^9)$,表示每个参赛选手的水平值.

输出描述 :

输出一个整数表示所有队伍的水平值总和最大值.

示例1 :

输入

```

2
5 2 8 5 1 5

```

输出

代码片段

功能实现			代码提交统计			代码执行统计		
	TA的	平均		TA的	平均	答案错误 : 2 答案正确 : 1		
总通过率	100%	50%	使用语言	C++				
基本测试用例通过率	6/6 (100%)	59%	做题用时	00:18:23	00:19:10			
边缘测试用例通过率	4/4 (100%)	37%	提交次数	3	4			
代码效率					代码规范及可读性			
	TA的	参考				代码规范得分		4.8
运行时间	178ms	1s				Line 8: Use int16/int64/etc, rather than the C type long [runtime/int] [4]		
占用内存	1272K	32768K						

他的代码：

做题用时: 18 分钟 语言: C++ 运行时间: 178ms 占用内存: 1272K 程序状态: 答案正确

```
#include<iostream>
#include<algorithm>
using namespace std;

int main()
{
    int num;
    long long sum = 0;
    cin >> num;
    int *a = new int[3 * num];
    for (int i = 0; i < 3 * num; i++)
    {
        cin >> a[i];
    }
    sort(a, a + 3 * num);
    int index = 3 * num - 2;
    while (num--)
    {
        sum += a[index];
        index -= 2;
    }
    cout << sum << endl;
    return 0;
}
```



点此或手机扫描二维码查看代码编写过程

23 [平均分7.3分 | 24人正确/154人做题 | 提交: 3 次 | 得分 : 18.0 / 20.0

标题：牛牛的数列 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

【牛牛的数列】牛牛现在有一个n个数组成的数列,牛牛现在想取一个连续的子序列,并且这个子序列还必须得满足:最多只改变一个数,就可以使得这个连续的子序列是一个严格上升的子序列,牛牛想知道这个连续子序列最长的长度是多少。

输入描述：

输入包括两行,第一行包括一个整数n($1 \leq n \leq 10^5$),即数列的长度;
第二行n个整数a_i, 表示数列中的每个数($1 \leq a_i \leq 10^9$),以空格分割。

输出描述：

输出一个整数,表示最长的长度。

示例1：

输入

6
7 2 3 1 5 6

输出

5

代码片段

功能实现			代码提交统计			代码执行统计	
总通过率	TA的 90%	平均 36%	使用语言	TA的 C++	平均	答案错误：3	
基本测试用例通过率	5/6 (83%)	39%	做题用时	00:36:48	00:25:11		
边缘测试用例通过率	4/4 (100%)	30%	提交次数	3	3		
代码效率					代码规范及可读性		
运行时间	TA的 57ms	参考 1s	代码规范得分 Line 28: If an else has a brace on one side, it should have it on both [readability/braces] [5]			4.80392	
占用内存	3176K	32768K					

他的代码：

做题用时: 36 分钟 语言：C++ 运行时间：57ms 占用内存：3176K 程序状态：答案错误

```
#include<iostream>
#include<algorithm>
using namespace std;
int MaxLong(int *a, int start, int n)
```

```

{
    if (n == 0)
        return 0;
    bool flag = false;
    int index;
    int temp;
    int count = 1;
    for (int i=start+1; i<n; i++)
    {
        if (a[i] <= a[i-1] && flag)
        {
            a[index] = temp;
            return max(count, MaxLong(a, index, n));
        }
        else if (a[i] <= a[i-1])
        {
            if (count == 1)
            {
                flag = true;
                index = i;
                temp = a[i];
                count++;
            }
            else{
                flag = true;
                index = i;
                temp = a[i];
                a[i] = a[i-1];
                count++;}
        }
        else
        {
            count++;
        }
    }
    return count;
}

int main()
{
    int num;
    cin >> num;
    int *a = new int[num];
    for (int i=0; i<num; i++)
        cin >> a[i];
    cout << MaxLong(a, 0, num);
    return 0;
}

```



[点此](#)或手机扫描二维码查看代码编写过程