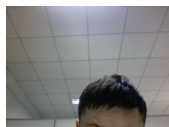


# Java方向每日一题day20\_1月5日-任栋-测评结果

## 考生信息



任栋

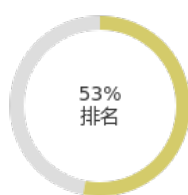
投递编号：2 | 学校：西安理工大学 | 邮箱：1104580363@qq.com | 职位：2020大四春招冲刺班 |

参考区域：陕西省西安市（221.11.20.102） |

做题用时：03:08:01(2021-01-05 21:10:24开始答题，2021-01-06 00:23:26交卷) | 作答设备：PC |

已同意诚信声明和隐私协议

## 考生成绩



题型	得分	正确题数	排名	用时	是否阅卷
单选	25.0	5	30	00:42:25	--
编程	50.0	2	1	02:20:35	--

## 知识点技能图谱

编程基础	1/1
树	1/2
递归	1/1
排序	0/1
模拟	1/1
堆	1/1
链表	1/2
栈	1/2
队列	1/1
哈希	1/1
图	0/1

知识点	得分	正确题数
编程基础	25.0	1
树	5.0	1
递归	5.0	1
排序	0.0	0
模拟	25.0	1
堆	5.0	1
链表	25.0	1
栈	5.0	1
队列	5.0	1
哈希	5.0	1
图	0.0	0

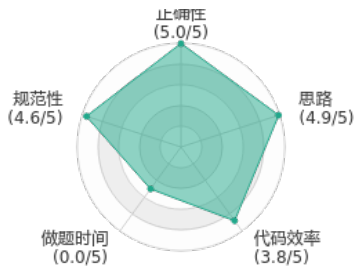
## 历史笔试记录

序号	试卷名称	排名	总得分	得分详情	作弊嫌疑	安排笔试时间	交卷时间
----	------	----	-----	------	------	--------	------

序号	试卷名称	排名	总得分	得分详情	作弊嫌疑	安排笔试时间	交卷时间
1	大四春招冲刺班JavaSE考试	31.0%	26.0/60	单选:26.0分	否	2020-10-28 17:33:16	2020-10-29 10:30:13
2	大四春招冲刺班数据结构考试	77.0%	22.0/60	单选:22.0分 编程:0.0分	否	2020-11-23 11:55:15	2020-11-24 10:40:01
3	Java方向每日一题day02_11月24日	70.0%	55.0/100	单选:30.0分 编程:25.0分	否	2020-11-23 12:10:19	2020-11-25 15:36:05
4	Java方向每日一题day03_11月25日	79.0%	55.0/100	单选:30.0分 编程:25.0分	否	2020-11-24 15:19:25	2020-11-24 22:21:11
5	Java方向每日一题day04_11月26日	61.0%	85.0/100	单选:35.0分 编程:50.0分	否	2020-11-25 14:58:20	2020-11-25 23:17:45
6	Java方向每日一题day05_11月27日	16.0%	85.0/100	单选:35.0分 编程:50.0分	否	2020-11-25 15:54:38	2020-11-26 23:32:02
7	Java方向每日一题day06_11月28日	43.0%	77.5/100	单选:30.0分 编程:47.5分	否	2020-11-27 14:19:26	2020-11-27 22:53:24
8	Java方向每日一题day07_11月30日	1.0%	100.0/100	单选:50.0分 编程:50.0分	否	2020-11-29 13:58:20	2020-11-30 22:17:40
9	Java方向每日一题day08_12月1日	28.0%	90.0/100	单选:40.0分 编程:50.0分	是，代码抄袭	2020-11-30 10:48:03	2020-12-01 22:29:01
10	Java方向每日一题day09_12月2日	86.0%	50.0/100	单选:35.0分 编程:15.0分	否	2020-12-01 10:43:40	2020-12-02 21:12:07
11	Java方向每日一题day10_12月3日	44.0%	75.0/100	单选:25.0分 编程:50.0分	否	2020-12-02 12:27:01	2020-12-04 09:14:04
12	Java方向每日一题day11_12月4日	72.0%	57.14/100	单选:25.0分 编程:32.14分	否	2020-12-03 10:46:54	2020-12-04 11:05:57
13	Java方向每日一题day12_12月5日	66.0%	60.0/100	单选:35.0分 编程:25.0分	否	2020-12-04 10:43:45	2020-12-05 21:50:32
14	每日一题Java方向day13_12月7日	75.0%	48.57/100	单选:40.0分 编程:8.57分	否	2020-12-05 10:31:45	2020-12-07 22:58:00
15	每日一题Java方向day14_12月8日	62.0%	65.0/100	单选:40.0分 编程:25.0分	否	2020-12-07 12:07:00	2020-12-08 16:49:09
16	每日一题Java方向day16_12月10日	34.0%	75.0/100	单选:25.0分 编程:50.0分	否	2020-12-09 10:58:00	2020-12-12 16:11:42
17	每日一题Java方向day17_12月11日	81.0%	45.0/100	单选:45.0分	否	2020-12-09 15:13:19	2020-12-13 16:22:36
18	Java方向每日一题day19_1月4日	83.0%	45.0/100	单选:20.0分 编程:25.0分	否	2021-01-03 18:27:10	2021-01-05 21:02:14

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
----	-----	------	------	------	------	------	------	------	------



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	01:24:21	Java	70ms	12884K			2%
编程题2	100%	12	00:56:14	Java	181ms	15784K			2%

1 [平均分2.9分 | 24人正确/41人做题 | 用时：<1分 | 得分：0.0 / 5.0]  
对于顺序存储的线性表，访问结点和增加结点的时间复杂度为（ ）。

- A  $O(n)$   $O(n)$
- B  $O(n)$   $O(1)$
- C  $O(1)$   $O(n)$
- D  $O(1)$   $O(1)$

他的回答：A (错误)

正确答案：C

2 [平均分2.4分 | 20人正确/42人做题 | 用时：<1分 | 得分：0.0 / 5.0]  
若栈采用顺序存储方式存储，现两栈共享空间 $V[1..m]$ ， $top[i]$ 代表第 $i$ 个栈( $i=1,2$ )栈顶，栈1的底在 $v[1]$ ，栈2的底在 $V[m]$ ，则栈满的条件是（ ）。

- A  $top[1]+top[2]=m$
- B  $top[1]+1=top[2]$
- C  $top[2]-top[1]=0$
- D  $top[1]=top[2]$

他的回答：A (错误)

正确答案：B

3 [平均分4.1分 | 34人正确/41人做题 | 用时：2分 | 得分：5.0 / 5.0]  
下述有关栈和队列的区别，说法错误的是？

- A 栈是限定只能在表的一端进行插入和删除操作。
- B 队列是限定只能在表的一端进行插入和在另一端进行删除操作。
- C 栈和队列都属于线性表
- D 栈的插入操作时间复杂度都是 $o(1)$ ，队列的插入操作时间复杂度是 $o(n)$

他的回答：D (正确)

正确答案：D

4 [平均分4.1分 | 34人正确/41人做题 | 用时：<1分 | 得分：5.0 / 5.0]

从前有座山，山里有座庙，庙里有个老和尚，再给小和尚讲故事，故事内容是：从前有座山，山里有座庙，庙里有个老和尚，再给小和尚讲故事，故事内容是：从前有座山，山里有座庙，庙里有个老和尚，再给小和尚讲故事，故事内容是.....描述的是()

- A 贪心
- B 回溯
- C 穷举
- D 分治
- E 递归

他的回答：E (正确)

正确答案：E

参考答案：  
递归指的是一个过程：函数不断引用自身，直到引用的对象已知。

- 5 [平均分2.6分 | 21人正确/41人做题 | 用时：19分 | 得分：0.0 / 5.0]
- 某二叉树共有 399 个结点，其中有 199 个度为 2 的结点，则该二叉树中的叶子结点数为（ ）
- A 不存在这样的二叉树
  - B 200
  - C 198
  - D 199

他的回答：D (错误)

正确答案：B

参考答案：

根据二叉树的基本性质，对任何一棵二叉树，度为 0 的结点（即叶子结点）总是比度为 2 的结点多一个。题目中度为 2 的结点为 199 个，则叶子结点为  $199+1=200$ 。故本题答案为 B 选项。

- 6 [平均分4.4分 | 36人正确/41人做题 | 用时：<1分 | 得分：5.0 / 5.0]
- 某二叉树的前序遍历序列与中序遍历序列相同，均为 ABCDEF，则按层次输出（同一层从左到右）的序列为（ ）
- A ABCDEF
  - B BCDEF A
  - C FEDCBA
  - D DEFABC

他的回答：A (正确)

正确答案：A

参考答案：

二叉树遍历可以分为 3 种：前序遍历（访问根结点在访问左子树和访问右子树之前）、中序遍历（访问根结点在访问左子树和访问右子树两者之间）、后序遍历（访问根结点在访问左子树和访问右子树之后）。二叉树的中序遍历序列和前序遍历序列均为 ABCDEF，可知该树只有右子树结点，没有左子树结点，A 为根结点。中序遍历序列与前序遍历序列相同说明该树只有右子树没有左子树，因此该树有 6 层，从顶向下从左向右依次为 ABCDEF。故本题答案为 A 选项。

- 7 [平均分3.0分 | 25人正确/41人做题 | 用时：15分 | 得分：5.0 / 5.0]
- 初始序列为1 8 6 2 5 4 7 3—组数采用堆排序，当建堆（小根堆）完毕时，堆所对应的二叉树中序遍历序列为：（ ）
- A 8 3 2 5 1 6 4 7
  - B 3 2 8 5 1 4 6 7
  - C 3 8 2 5 1 6 7 4
  - D 8 2 3 5 1 4 7 6

他的回答：A (正确)

正确答案：A

参考答案：

解析：堆排序：利用堆的性质进行的一种选择排序 答案：A

- 8 [平均分4.1分 | 34人正确/41人做题 | 用时：<1分 | 得分：5.0 / 5.0]

解决散列法中出现冲突问题常采用的方法是\_\_\_\_\_。

- A 数字分析法、除余法、平方取中法
- B 数字分析法、除余法、线性探测法
- C 数字分析法、线性探测法、多重散列法
- D 线性探测法、多重散列法、链地址法

他的回答：D (正确)

正确答案：D

9 [平均分2.9分 | 24人正确/42人做题 | 用时 : <1分 | 得分 : 0.0 / 5.0

以下哪种排序算法对(1,3,2,4,5,6,7,8,9)进行的排序最快？

- A 冒泡
- B 快排
- C 归并
- D 堆排

他的回答 : B (错误)

正确答案 : A

10 [平均分3.5分 | 29人正确/41人做题 | 用时 : <1分 | 得分 : 0.0 / 5.0

设无向图的顶点个数为n，则该图最多有多少条边？

- A n-1
- B  $n(n+1)/2$
- C  $n(n-1)/2$
- D n
- E 不同于以上答案

他的回答 : E (错误)

正确答案 : C

11 完善核心代码 | 语言限制 [平均分23.2分 | 38人正确/41人做题 | 提交: 1 次 | 得分 : 25.0 / 25.0

标题：微信红包 | 时间限制：3秒 | 内存限制：32768K | 语言限制：[Python, C++, C#, Java]

【微信红包】

春节期间小明使用微信收到很多个红包，非常开心。在查看领取红包记录时发现，某个红包金额出现的次数超过了红包总数的一半。请帮小明找到该红包金额。写出具体算法思路和代码实现，要求算法尽可能高效。

给定一个红包的金额数组gifts及它的大小n，请返回所求红包的金额。

若没有金额超过总数的一半，返回0。

测试样例：

[1,2,3,2,2],5

返回：2

输入描述：

输出描述：

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	平均		TA的	平均	答案正确：1	
	100%	92%	使用语言	Java			
	1/1 (100%)	92%	做题用时	01:24:21	00:21:32		
基本测试用例通过率			提交次数	1	3		
代码效率					代码规范及可读性		
	TA的	参考				代码规范得分	
运行时间	70ms	3s				4.8	
占用内存	12884K	32768K				Line 4:42: Parameter name 'n' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*\$'. [ParameterName]	

他的代码：

```
import java.util.*;

public class Gift {
    public int getValue(int[] gifts, int n) {
        // write code here
        HashMap<Integer,Integer> hashMap = new HashMap<>();
        for (int i = 0; i < gifts.length; i++) {
            if (!hashMap.containsKey(gifts[i])){
                hashMap.put(gifts[i],1);
            }else {
                hashMap.put(gifts[i],hashMap.get(gifts[i])+1);
                if (hashMap.get(gifts[i]) > n/2){
                    return gifts[i];
                }
            }
        }
        return 0;
    }
}
```



[点此](#)或手机扫描二维码查看代码编写过程

12

完善核心代码

语言限制

[平均分22.3分 | 33人正确/37人做题 | 提交: 12 次 | 得分 : 25.0 / 25.0]

标题：链表分割 | 时间限制：3秒 | 内存限制：32768K | 语言限制：[Python, C++, C#, Java]

【链表分割】

编写代码，以给定值x为基准将链表分割成两部分，所有小于x的结点排在大于或等于x的结点之前  
给定一个链表的头指针 ListNode\* pHead，请返回重新排列后的链表的头指针。注意：分割以后保持原来的数据顺序不变。

输入描述：

输出描述：

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	平均	使用语言	TA的	平均	答案错误	: 10		
	100%	89%		Java			运行超时	: 1	
基本测试用例通过率	1/1	89%	做题用时	00:56:14	00:16:57	答案正确	: 1		
	(100%)		提交次数	12	4				
代码效率					代码规范及可读性				

TA的	参考	代码规范得分	4.4
运行时间	181ms	3s	
占用内存	15784K	32768K	
		Line 13:40: Parameter name 'pHead' must match pattern '^ [a-z][a-z0-9][a-zA-Z0-9]*\$'. [ParameterName]	
		Line 13:51: Parameter name 'x' must match pattern '^ [a-z][a-z0-9][a-zA-Z0-9]*\$'. [ParameterName]	
		Line 18: Distance between variable 'smaller' declaration and its first usage is 4, but allowed 3. Consider to make that variable as final if you still need to store its value in advance (before method calls that might do side effect on original value). [VariableDeclarationUsageDistance]	

他的代码：

做题用时: 56 分钟    语言 : Java    运行时间 : 181ms    占用内存 : 15784K    程序状态 : 答案正确

```
import java.util.*;

/*
public class ListNode {
    int val;
    ListNode next = null;

    ListNode(int val) {
        this.val = val;
    }
}*/
public class Partition {
    public ListNode partition(ListNode pHead, int x) {
        // write code here
        ListNode tmp = pHead;
        ListNode small = new ListNode(0);
        ListNode large = new ListNode(0);
        ListNode smaller = small;
        ListNode greater = large;
        while (tmp != null){
            if (tmp.val < x){
                small.next = tmp;
                small = small.next;
            }else {
                large.next = tmp;
                large = large.next;
            }
            tmp = tmp.next;
        }
        large.next = null;
        small.next = greater.next;
        return smaller.next;
    }
}
```



[点此](#)或手机扫描二维码查看代码编写过程