

2020春季数据结构与算法模拟笔试

Updated 1536 GMT+8 May 21, 2024

2024 spring, Compiled by Hongfei Yan

Logs:

2024/5/19, 题目来自“笔试样题2”，我没有标准答案，现有答案是自己做的。

一. 选择题 (14)

1. 一个深度为 h 的满 k 叉树，最多有多少个结点？（独根树深度为0）（）

A、 $\frac{K^{h+1}-1}{K-1}$

B、 K^{h-1}

C、 K^h

D、 $\frac{K^h-1}{K-1}$

2. 对二叉排序树（即BST，也称“二叉搜索树”）进行什么遍历，可以得到该二叉树所有结点构成的排序序列？（）

A、前序 preorder

B、后序 postorder

C、按层次 levelorder

D、中序 inorder

3. 2-3树是一种特殊的树，它满足两个条件：

(1) 每个内部结点有两个或三个子结点；(2) 所有的叶结点到根的路径长度相同；

如果一棵2-3树有10个叶结点，那么它可能有_____个非叶结点。（多选）

A、8

B、7

C、5

D、6

4. 用相邻矩阵 A 表示图，判定任意两个顶点 V_i 和 V_j 之间是否有长度为 m 的路径相连，则只要检查_____的第 i 行第 j 列的元素是否为零即可。

A、 A^m

B、 mA

C、A

D、 A^{m-1}

5. 有一严格升序的整型数组A，元素个数为n。现将其前k ($0 \leq k \leq n$) 个元素整体移动到数组后面，得到数组B，使B数组的前n-k个元素恰好是A数组的后n-k个元素，B数组的后k个元素恰好是A数组的前k个元素，且前后两部分的内部升序仍保持不变。请设计一个算法在B数组中查找某个给定元素value。算法设计在函数searchValue中，函数头可采用searchValue(int B[], int value)。那么你设计的高效算法的时间复杂度是 _____

A、 $O(n)$

B、 $O(n^{0.5})$

C、 $O(\log n)$

D、 $O((\log n)^2)$

6. 对于排序算法特性的叙述正确的是（多选）（）

A、冒泡排序不需要访问那些已排好序的记录

B、shell排序过程中，当对确定规模的这些小序列进行插入排序时，要访问序列中的所有记录

C、选择排序需要访问那些已排好序的记录

D、快速排序过程中，递归树上根据深度划分的每个层次都要访问序列中的所有记录

E、归并排序过程中，递归树上每个层次的归并操作不需要访问序列中的所有记录

7. 排序算法大都是基于数组实现的，大部分的算法也能用链表来实现，但有些特殊的算法不适合线性链表存储，不适合（使算法复杂度增大）链式存储的算法有（多选）（）

A、直接选择排序 Straight Selection Sorting

B、插入排序 Insertion sorting

C、堆排序 Heap sorting

D、shell排序 Shell sorting

8. 假定有k个关键码互为同义词，即hash函数结果相同，若采用闭散列的线性探测法把这k个关键字存入散列表中，至少要进行_____次探测？

A、k

B、k-1

C、 $k*(k+1)/2$

D、k+1

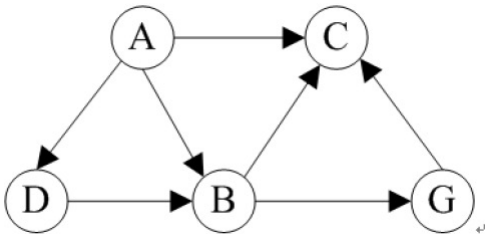
9. AVL树中任何节点的两个子树的高度最大差别为____ AVL树查找时间为 $O()$ ，**树的结构**(会/不会)改变（）

A、 $1 \log N$ 不会will not

- B、 $0 \log N$ 会will
- C、 $1 N$ 不会will not
- D、 $1 \log N$ 会will

#D

10. 设栈S 和队列Q 的初始状态为空，元素a, b, c, d, e依次通过栈S，一个元素出栈后即进队列Q，若5个元素出队的序列是c, e, d, b, a则栈S的容量至少应该是_____。
- A、 2
 - B、 3
 - C、 4
 - D、 5
11. 下列关于堆的说法正确的有（多选） （ ）
- A、 最小堆中，最下面一层最靠右的结点一定是权值最大的结点。
 - B、 堆一定是完全二叉树。
 - C、 最小堆中，某个结点左子树中最大的结点可能比右子树中最小的结点小。
 - D、 使用“下沉”法建堆要比将元素一个一个插入堆来建堆效率高。
12. 如图所示的DAG图，其拓扑排序序列为_____。



- A、 ADGBC
 - B、 ADBGC
 - C、 AGBDC
 - D、 ACDGB
13. 下列关于最短路算法的说法正确的有（多选） （ ）
- A、 当图中不存在负权回路但是存在负权边时，Dijkstra算法不一定能求出源点到所有点的最短路。
 - B、 当图中存在负权回路时，Dijkstra算法也一定能求出源点到所有点的最短路。
 - C、 当图中不存在负权边时，Dijkstra算法能求出每对顶点间最短路径。
 - D、 Dijkstra算法不能用于每对顶点间最短路计算。

14、顺序栈是用一段连续的空间存储内容，本质是顺序表。链式栈则是采用单链表的方式存储。下列关于这两种存储方式的说法正确的是（多选）（）

- A、顺序栈的压栈和出栈操作只需常数时间。
- B、链式栈的压栈和出栈操作只需常数时间。
- C、顺序栈需要指定一个具体的长度
- D、链式栈需要一个结构性开销

二. 填空 (16)

1. 由小到大写出以下时间复杂度的序列：

(1) n^2+100n

(2) $3n^2+100n^2$

(3) $10+3\log_{10}n$

(4) $10n+20n\log_{10}n$

(5) $2n$

(6) $1000n$

答案直接写标号，如：(1)(2)(3)(4)(5)

2. 计算运行下列程序段后s的值：

```
n = 10; s = 0;
for k in range(1, n-1):
    for j in range(n, k-1, -1):
        s += 1
```

#52

3、双端队列可以在队列的两端进行插入和删除操作，既可在队尾进行插入/删除，又可在队头进行插入/删除。现有11个不同的元素顺序输入到双端队列，那么可以得到多少种不同的排列？

4. 使用KMP算法求出模式p="aabcaabbbaa"的next数组。注意：只列出数字，数字之间用一个空格分隔。比如：0 0 0 0 0 0 0 0 0

0 1 0 0 1 2 3 0 1 2

5. 利用上题p="aabcaabbbaa"的Next数组，对t="aaabaabcabaabcaabbaab"进行匹配。有多少次字符比较？（注意：每一次p中的字符与t中的字符的一次比较计做一次）

6. 一个有4层结点的完全二叉树。按前序遍历周游给结点从1开始编号，则第21号结点的父结点是多少号？（注释：根的层数为0）

7.假设一棵二叉树中，度为2的结点有20个，度为1的结点有10个，度为0的结点有多少个？

8.某二叉树中序序列为A,B,C,D,E,F,G, 前序序列为E,A,C,B,D,G,F, 则后序序列是？

(注意：答案不要含空格和逗号，比如可以是ABCDEFG)

9.对于键值序列{38,64,52,26,73,40,48,55,15,12}，用筛选法建最小值堆，共交换元素多少次？

#要用筛选法建立最小值堆，需要进行一系列的比较和交换操作，直到满足最小堆的性质。下面是建立最小值堆的具体步骤：

1. 从最后一个非叶子节点开始，向前遍历到根节点，对每个节点执行下沉操作。
2. 下沉操作：比较当前节点与其子节点的值，若当前节点的值大于其子节点的值，则将其与较小的子节点交换，直到满足最小堆的性质。

首先，初始化该键值序列：{38, 64, 52, 26, 73, 40, 48, 55, 15, 12}

现在开始建立最小值堆：

1. 从最后一个非叶子节点开始，该节点索引为 $(n/2 - 1)$ ，其中 n 为序列长度，即 $10/2 - 1 = 4$ 。开始下沉操作。
 2. 对于节点索引为 4 (值为 73)，与其子节点比较并交换，依次向上直到根节点。
 - 73 与 15 交换：{38, 64, 52, 26, 15, 40, 48, 55, 73, 12}
 - 15 与 52 交换：{38, 64, 15, 26, 52, 40, 48, 55, 73, 12}
 - 15 与 26 交换：{38, 15, 64, 26, 52, 40, 48, 55, 73, 12}
 - 15 与 38 交换：{15, 38, 64, 26, 52, 40, 48, 55, 73, 12}
3. 继续下一个节点，索引为 3 (值为 26)，进行下沉操作。
 - 26 与 12 交换：{15, 12, 64, 38, 52, 40, 48, 55, 73, 26}
4. 继续下一个节点，索引为 2 (值为 64)，进行下沉操作。
 - 64 与 48 交换：{15, 12, 48, 38, 52, 40, 64, 55, 73, 26}
5. 继续下一个节点，索引为 1 (值为 12)，进行下沉操作。
 - 12 与 38 交换：{15, 12, 48, 26, 52, 40, 64, 55, 73, 38}
 - 12 与 15 交换：{12, 15, 48, 26, 52, 40, 64, 55, 73, 38}
6. 继续下一个节点，索引为 0 (值为 12)，进行下沉操作。

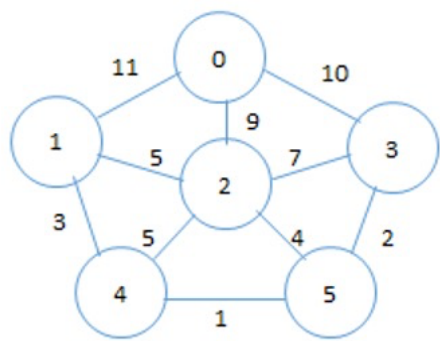
现在，最小值堆建立完成，序列为：{12, 15, 48, 26, 52, 40, 64, 55, 73, 38}。

交换元素的总次数为 8 次。

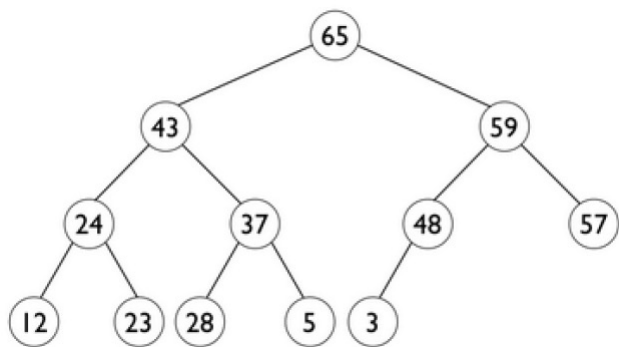
10.从空二叉树开始，严格按照二叉搜索树的插入算法（不进行旋转平衡），逐个插入关键码{15, 82, 10, 4, 55, 89, 29, 45, 54, 35, 25}构造出一颗二叉搜索树，对该二叉搜索树按照后序遍历得到的序列为（每两个元素之间用一个空格隔开）

11. 请使用Kruskal算法求出下图的最小生成树，依次写出每次被选择的合法的合并代价最小的边的编号，用一个空格分隔（如果同时存在多条边满足要求，选择编号最小的）。顶点a到顶点b ($a < b$) 之间的边编

号为ab，例如图中权值为1的边编号为45。



12. 有一组待排序的记录，其排序码为{18, 5, 20, 30, 9, 27, 6, 14, 45, 22}，而采用直接选择排序的比较次数是
13. 将序列(p, h, n, d, y, a, f, q, x, m, c, e) 中的关键码按字母升序重新排序，初始步长为4的shell排序一趟扫描的结果为（用一个空格分隔字母）
14. 现有中缀表达式 $E = ((20 + 5) / 5 + 5 * (30 - 7)) * 4$ ，与E等价的后缀表达式为
15. 一棵完全二叉树的第6层上有23个叶子结点，则此二叉树最多有 () 个结点。（独根树高度为1)
- 16.对于如下图所示的最大堆，插入元素70后，堆的后序遍历结果是



三. 简答题（5）

1. 请比较有序表Ordered List 和二叉搜索树BST 两种数据结构，并分析其put/get 操作的异同？
2. 什么是强连通分支？强连通分支算法所用到的转置图Transpose Graph 的定义是什么？
3. 请比较Dijkstra 算法和Prim 算法的异同？
4. 双端队列Deque 可以用两个栈 Stack 来实现么？为什么？
- #是的，双端队列（Deque，即Double-Ended Queue）可以使用两个栈（Stack）来实现。
- 双端队列是一种具有队列和栈性质的数据结构，允许在队列的两端进行插入和删除操作。栈是一种后进先出（LIFO）的数据结构，而队列是一种先进先出（FIFO）的数据结构。尽管栈和队列的性质不同，但是可以利用两个栈的特性来模拟实现双端队列。
- 具体实现方法如下：

- 1. 使用两个栈，分别称为 `stack1` 和 `stack2`。
- 2. 元素的插入操作 (`push_back`) 时，将元素压入 `stack1`。
- 3. 元素的删除操作 (`pop_front`) 时，首先检查 `stack2` 是否为空。如果不为空，直接从 `stack2` 弹出栈顶元素；如果为空，将 `stack1` 中的所有元素逐个弹出并压入 `stack2`，然后从 `stack2` 弹出栈顶元素。
- 4. 元素的插入操作 (`push_front`) 时，将元素压入 `stack2`。
- 5. 元素的删除操作 (`pop_back`) 时，首先检查 `stack1` 是否为空。如果不为空，直接从 `stack1` 弹出栈顶元素；如果为空，将 `stack2` 中的所有元素逐个弹出并压入 `stack1`，然后从 `stack1` 弹出栈顶元素。

通过以上操作，我们可以使用两个栈模拟实现双端队列的插入和删除操作。这是因为栈的特性使得元素在插入和删除时可以保持正确的顺序。

需要注意的是，使用两个栈实现的双端队列可能会涉及到元素的频繁移动，因此在某些情况下可能不如直接使用链表或动态数组实现的双端队列高效。但从概念上来说，使用两个栈来实现双端队列是可行的。

- 5. 请描述AVL树插入时的四种旋转情况以及如何旋转，可以画图

四. 综合题 (4)

1. 定义函数fab(n)为Fabonacci 数列第n个数，其中fab(1)=1，请：

- 1) 用递归方法写出fab(n)的算法，并分析fab(n)算法的时间复杂度；
- 2) 观察到上一问计算过程中，有很多中间结果被重复计算，请改进（1）中算法得到新算法fab2(n)，并分析fab2(n)算法的时间复杂度。
- 2. 将1、2、3...n这n个整数依次入栈，期间可以出栈，直到栈空，这样会得到一个出栈序列。例如4个数，如果1入、2入、3入、3出、4入、4出、2出、1出，则出栈序列为3、4、2、1。合法的出栈序列会有多个，请设计一个算法，判断给定的一个包含n个整数的序列是否是合法的出栈序列。
- 3. 请编写算法maxsub，返回两个字符串astr和bstr的1个最长公共子串，给定astr的长度不大于bstr。例如：maxsub("kabcd", "nbcyuu")返回"bc"，maxsub("123", "abcd")返回""。
- 4. 给定下面的树，不用考虑节点上的数字，按字母大小排序
- 1) 逐步图示并说明经过怎样合适的旋转可以让它变得平衡；
- 2) 平衡后的树作为普通BST 构建的话，请给出节点的插入过程（答案不唯一）。

