

上海科技大学 2022 年攻读硕士学位研究生
招生考试试题

科目代码：991

科目名称：数据结构与算法

考生须知：

1. 本试卷满分为 150 分，全部考试时间总计 180 分钟。
 2. 所有答案必须写在答题纸上，写在试题纸上或草稿纸上均无效。
 3. 考生可用中文或者英文作答。
-

一、判断题（10 题，每题 2 分，共 20 分）

请在答题纸上写明下列每个命题的真假。真则打“√”，假则打“×”。

- 1) 顺序存储结构要求连续的存储空间，在存储管理上不够灵活，因此不常用。（ ）
- 2) 访问链式存储方式的线性表的第 k 个元素的时间复杂度与 k 的大小有关。（ ）
- 3) 求解最小生成树（Minimum Spanning Tree）的两种算法，即克鲁斯卡尔（Kruskal）算法和普里姆（Prim）算法均属于贪心算法。（ ）
- 4) 假设堆栈（Stack）的输入序列是 $1, 2, \dots, n$ ，输出序列是 a_1, a_2, \dots, a_n ，如果已知 $a_i = n (1 \leq i \leq n)$ ，则有 $a_i > a_{i+1} > \dots > a_n$ 。（ ）
- 5) 快速排序（Quick Sort）在任何情况下的时间复杂度都是 $O(n \log n)$ 。（ ）
- 6) 已知一个数组 A ， $A[0] = \text{null}$ ，且 $A[1] \sim A[n]$ 依次增大，则数组 A 同时也可以认为是一个二叉最小堆的数组存储。（ ）
- 7) 二分搜索树（Binary Search Tree, BST）中，最大的节点一定是叶子节点。（ ）
- 8) 如果两个二叉树的先序遍历完全一致，且后序遍历也完全一致，则两个二叉树一定为同一个二叉树。（ ）
- 9) 以下检查输入数 n 是否为质数的算法以多项式时间运行：对于每个整数 $2 \leq i \leq \sqrt{n}$ ，检查 i 是否整除 n ，如果是，则声明 n 不是质数。如果不存在这样的 i ，则声明 n 是质数。（ ）
- 10) 给定问题 Y ，如果 NP 中任何一个问题 X 都可以多项式时间归约到问题 $Y (X \leq_p Y)$ ，那么问题 Y 一定是 NPC 问题（NP-Complete）。（ ）

二、单选题（15 题，每题 2 分，共 30 分）

每题只有一个正确选项。请在答题纸上写下正确选项的序号。

- 1) 以下关于渐进记号 (Asymptotic Analysis) 的等式错误的是: ()。
- A. $O(n) + O(n^4) = O(n^4)$
B. $O(n) + \Theta(n^2) = \Theta(n^2)$
C. $O(n^2) + \Theta(n^2) = \Theta(n^2)$
D. $O(n^2) + \Theta(n^2) = O(n^2)$
- 2) 考虑一个从一个包含 n 个整数的数组里面删除所有负值的算法。算法顺序地检查数组里面的每个元素, 当发现负值时, 删除该负值, 并将其后的每个元素前移一位。这个算法的复杂度是: ()。
- A. $O(n)$ B. $\Theta(n)$ C. $O(n^2)$ D. $\Theta(n^2)$
- 3) 执行 () 操作时, 需要使用队列 (Queue) 作为辅助存储空间?
- A. 广度优先 (Breadth-First Traversal) 遍历二叉树
B. 深度优先 (Depth-First Traversal) 遍历二叉树
C. 查找哈希表 (Hash Table)
D. 先序遍历 (Pre-Order Traversal) 二叉树

- 4) 一个长度为 10 的哈希表/散列表 (Hash Table) 使用哈希函数 (Hash Function): $h(k) = k \bmod 10$ 。当冲突发生时, 使用开放寻址方式中的线性寻址法解决冲突。在空表中插入 6 个键值之后, 表中元素如表 1 中所示:

索引 (Index)	0	1	2	3	4	5	6	7	8	9
键值 (Value)	60	71	2	100	94	31				

表 1: 题 4) 中所述哈希函数和寻址方式产生的哈希表

下列哪个选项给出了键值依次插入表中的可能顺序? ()。

- A. 60, 71, 2, 100, 31, 94
B. 60, 94, 31, 100, 2, 71
C. 2, 60, 100, 71, 94, 31
D. 94, 71, 2, 60, 100, 31
- 5) 在哈希表 (Hash Table) 中, 哈希冲突 (Collision) 指的是: ()。
- A. 两个意义相近的单词
B. 具有相同哈希地址的两个元素
C. 被映射到不同哈希地址的一个元素
D. 被不同哈希函数映射到同一地址的两个元素
- 6) 假设分层文件系统中的路径信息以树的方式存储, 其中任意目录的子目录为该目录对应节点的子节点。若需要将路径信息以下图的顺序自上而下打印出来, 应该采用以下哪种顺序对树进行遍历: ()。

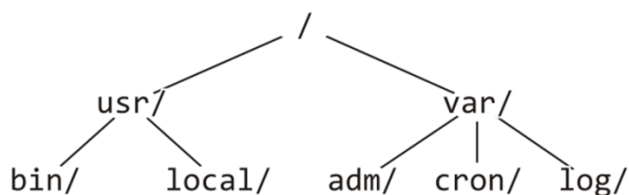


图 1: 题 6) 中所述分层路径系统

- A. 深度优先-先序遍历 (Depth-First Traversal, DFS Pre-Order)
 B. 深度优先-中序遍历 (Depth-First Traversal, DFS In-Order)
 C. 深度优先-后序遍历 (Depth-First Traversal, DFS Post-Order)
 D. 广度优先 (Breadth-First Traversal, BFS)
- 7) 一个完美二叉树有 n 个非叶子节点, 则该二叉树总共有 () 个节点。
 A. $n+1$
 B. $n+4$
 C. $2n+2$
 D. $2n+1$
- 8) 以下结论中哪一项是错误的? ()
 A. 完美二叉树 (Perfect Binary Tree) 一定是 AVL 平衡二叉树 (AVL Tree)
 B. 完全二叉树 (Complete Binary Tree) 一定是满二叉树 (Full Binary Tree)
 C. 完美二叉树 (Perfect Binary Tree) 一定是完全二叉树 (Complete Binary Tree)
 D. 完全二叉树 (Complete Binary Tree) 一定是 AVL 平衡二叉树 (AVL Tree)
- 9) 以下结论中哪一项是错误的? ()
 A. 插入操作在 AVL 平衡二叉树 (AVL Tree) 中的时间复杂度是 $O(\log n)$
 B. 查找操作在完全二叉搜索树 (Complete Binary Search Tree) 中的时间复杂度为 $O(n \log n)$
 C. 查找操作在高度为 h 的二叉搜索树 (Binary Search Tree) 中的时间复杂度为 $O(h)$
 D. 插入操作在有 n 个节点的二叉搜索树中的时间复杂度为 $O(n)$
- 10) 下面哪种排序算法在最好情况下的 (Best Case) 时间复杂度最低? ()
 A. 选择排序
 B. 冒泡排序
 C. 快速排序
 D. 插入排序
- 11) 假设你手边有台计算机使用选择排序对 400 个数字排序花了 400ms, 如果花费 1600ms, 大概能对多少个数字进行排序? ()
 A. 800
 B. 1200
 C. 1600
 D. 3200

- 12) 当使用“递归树 (Recursion tree)”来求解归并排序 (Merge Sort) 的时间复杂度时, 这个“递归树”大概有 () 层 (作为 n 的函数, n 是输入数组的长度)?
- A. 一个常数(与 n 无关)
 B. \sqrt{n}
 C. $\log_2(n)$
 D. n
- 13) 以下哪个选项有在多项式时间内求解的算法? ()
- A. 确定布尔表达式是否存在可满足的赋值
 B. 确定一个图是否存在哈密顿回路
 C. 确定一个整数集合是否具有总和为某个值 t 的子集
 D. 确定一个有向无环图中的最长路径
- 14) 大数乘法是利用 () 实现的算法?
- A. 分治策略 (Divide and Conquer)
 B. 动态规划 (Dynamic Programming)
 C. 贪心法 (Greedy)
 D. 回溯法 (Backtracking)
- 15) 如图 2 所示的无向图中, 如果使用克鲁斯卡尔 (Kruskal) 算法计算最小生成树 (Minimum Spanning Tree), 那么最后一个被选中的边是 () ?

- A. (c, e)
 B. (d, e)
 C. (b, g)
 D. (a, b)

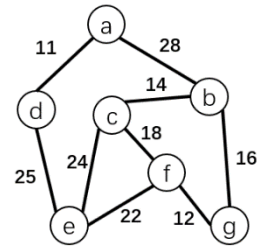


图 2: 题 15) 中所述图

三、简答题 (8 题, 共 35 分)

请在答题纸上简要回答以下问题。

- 1) 对于每对给出的函数 $f(n)$ 和 $g(n)$, 判断两个函数之间正确的渐进记号关系 (Asymptotic Analysis), 例如 $f(n) = O(g(n))$ 、 $f(n) = \Theta(g(n))$ 或者 $f(n) = \Omega(g(n))$ 。这些关系中的零个、一个或多个可能成立, 列出所有正确的关系, 并证明你的答案是正确的。(6 分)
- a) $f(n) = n^3 + n \log n$; $g(n) = n^3 + n^2 \log n$.
 b) $f(n) = \log \sqrt{n}$; $g(n) = \sqrt{\log n}$.
 c) $f(n) = \log(2^n)$; $g(n) = \log(3^n)$
- 2) 请简述分治算法 (Divide-and-Conquer) 与动态规划 (Dynamic Programming) 的异同点。(4 分)
- 3) 什么是循环队列 (Circular Array)? (3 分)
- 4) 递归算法 (Recursive Algorithm) 和非递归算法比较有哪些主要的优点和缺点? (4 分)
- 5) 推导出 $T(n)$ 的渐近上下界 (asymptotic upper/lower bounds), 使所给出的界尽量紧确 (make your bounds as tight as possible)。你可以假设 n 是 2 的幂。(6 分)

$$T(1) = 1$$

$$T(n) = T\left(\frac{n}{2}\right) + \log_2 n, \text{ if } n > 1$$

- 6) 请简述如何在有向图中检测环路, 以及该算法的时间复杂度? (4分)
- 7) 请简述如何在有向图中确定是否存在从节点 s 到节点 t 的通路, 并计算该算法的时间复杂度。(4分)
- 8) 请为数列 6, 5, 10, 9, 20, 15, 18 画一个高度最小的二分查找树 (Binary Search Tree)。(4分)

四、排序算法 (4 题, 每题 2.5 分, 共 10 分) 在本题中我们针对同一个输入序列分别执行不同的排序算法, 并将每种排序算法的部分输出序列在小题 (1) ~ (4) 中分别列出。请注意, 这些步骤不一定表示该算法中连续的输出步骤 (即可能缺少多个步骤), 但它们的输出顺序正确。对于每个小问, 请从下列选项中选择正确的排序算法选项: 插入排序 (Insertion Sort)、选择排序 (Selection Sort)、归并排序 (Merge Sort)、快速排序 (Quick Sort) 和堆排序 (Heap Sort); 并请说明你得出相应结论的理由。

输入序列: 29, 60, 82, 27, 15, 20, 65, 90, 69, 10, 24

- 1) 29, 60, 82, 15, 27, 20, 65, 90, 69, 10, 24
29, 60, 15, 27, 82, 20, 65, 90, 10, 24, 69
15, 27, 29, 60, 82, 10, 20, 24, 65, 69, 90
- 2) 27, 15, 20, 10, 24, 29, 60, 82, 65, 90, 69
15, 20, 10, 24, 27, 29, 60, 82, 65, 90, 69
10, 15, 20, 24, 27, 29, 60, 65, 90, 69, 82
- 3) 27, 29, 60, 82, 15, 20, 65, 90, 69, 10, 24
15, 27, 29, 60, 82, 20, 65, 90, 69, 10, 24
15, 20, 27, 29, 60, 82, 65, 90, 69, 10, 24
- 4) 29, 60, 82, 90, 24, 20, 65, 27, 69, 10, 15
82, 69, 65, 60, 24, 20, 15, 27, 29, 10, 90
10, 69, 65, 60, 24, 20, 15, 27, 29, 82, 90

五、AVL 平衡二叉树 (AVL Binary Search Tree) (3 题, 共 15 分): 请将数列以 35, 47, 18, 26, 15, 10, 30 的顺序依次插入一个初始为空节点的 AVL 平衡二叉搜索树。

- 1) 请画出所有节点插入结束后的 AVL 平衡二叉搜索树。(5分)
- 2) 请画出移除节点 47 后的 AVL 平衡二叉搜索树。(5分)
- 3) 请写出并证明在一个树高为 h 的 AVL 平衡二叉树中完成一次移除结点操作的时间复杂度。(5分)

六、算法设计 (3 题, 共 20 分): 假设算法的给定输入包含一家公司的利润历史数据, 即每一年赚钱及亏本信息, 设计目标是找出该公司在哪一个年段赚的钱最多。一个例子如下表所示, 答案是从第 5 年到第 8 年 (如表 3 红框所示), 该公司赚的钱最多, 为 9 百万人民币。

年份	1	2	3	4	5	6	7	8	9
利润(百万元人民币)	3	2	1	-7	5	2	-1	3	-1

表 3: 题 6 中公司利润历史数据

我们给出如下一个正式的数学定义:

输入: 一个包括有正有负的数组: $A[1 \dots n]$;

输出: 找到最大的 $V(i, j)$ 及对应的区间 $[i, j]$, $V(i, j)$ 的定义是

$$V(i, j) = \sum_{k=i}^j A[k]$$

- 1) 如果使用暴力法 (Brute-Force) 进行求解, 请给出该算法过程, 并分析它的时间复杂度。(6 分)
- 2) 如果使用分治法 (Divide-and-Conquer) 进行求解, 请给出该算法过程, 并分析它的时间复杂度。(7 分)
- 3) 如果使用动态规划 (Dynamic Programming) 进行求解, 请给出该算法过程, 并分析它的时间复杂度。(7 分)

七、NPC (1 题, 共 20 分) 我们先回顾一下 NP-Complete 问题 SUBSET-SUM: 给定一组非负整数 S 和一个目标值 K , 询问 S 是否存在一个子集 S' , 使得 $\sum_{i \in S'} i = K$ 。现在我们定义另外一个问题 AVERAGE-SUM, 即给定一组非负整数 S , 询问 S 是否存在一个子集 S' , 使得 $\sum_{i \in S'} i = \frac{1}{|S|} \sum_{i \in S} i$, $|S|$ 是 S 中元素的个数。它类似于 SUBSET-SUM 问题, 不同之处

在于现在的目标值总是 S 的平均值。请证明 AVERAGE-SUM 是 NP-Complete 问题。

提示: 证明问题 Y 是 NP-Complete 问题时, 需要首先证明该问题是 NP 问题, 然后再选择一个 NP-Complete 问题 X , 证明该问题 X 可以以多项式时间归约到问题 Y ($X \leq_p Y$)