

上海科技大学 2021 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一律无效。
 3. 考生可用中文或者英文作答。
-

1. 判断题（10 题，每题 1 分，共 10 分）

请在答题纸上写明题号后，对正确的命题则打“√”，错误的命题则打“×”。

- 1) 数组（Array）是一种线性结构，因此只能用来存储线性表。（ ）
- 2) 设堆栈 S 和队列 Q 的初始状态为空，元素 a1, a2, a3, a4, a5, a6 依次通过堆栈 S，一个元素出栈后即进入队列 Q，如果 6 个元素出队列 Q 的顺序是 a2, a4, a3, a6, a5, a1, 那么堆栈 S 的容量至少是 3。（ ）
- 3) 长度为 n 的单向链表（Linked list），如果查找每个元素的概率相同，则查找表中任意元素的平均查找长度为 $\Theta(n)$ 。（ ）
- 4) 冒泡排序（Bubble sort）算法中的比较次数与初始元素序列的排列无关。（ ）
- 5) 在排序算法中，快速排序（Quick sort）的执行时间一定最短。（ ）
- 6) 一个复杂度为 $T(n) = 4n^3 + n^2 \log_2(n)$ 的算法执行时间一定比一个复杂度为 $T(n) = n^3 + 6 \log_2(n)$ 的算法执行时间长。（ ）
- 7) 一棵完全二叉树（Complete binary tree）的节点数量为 n，则该树中节点的平均深度为 $\Theta(\log_2(n))$ 。（ ）
- 8) 如果要存储一个稀疏图，邻接矩阵法的空间复杂度小于邻接表法。（ ）
- 9) 寻找最短路径的迪杰斯特拉（Dijkstra）算法不属于贪心法。（ ）
- 10) 所有的 NP 问题都可以约化到 NPC 问题。（ ）

2. 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) 下列数据结构中, 哪一个**不是**线性结构? ()。
- A. 链表
 - B. 完全二叉树
 - C. 堆栈
 - D. 循环队列 (Circular queue)
- 2) 设计一个可以用于判别表达式中左、右括号是否配对出现的算法, 采用 () 数据结构最有效。
- A. 线性表的顺序存储结构
 - B. 队列
 - C. 堆栈
 - D. 线性表的链式存储结构
- 3) 下列算法中有哪个**不是**利用了分治算法实现? ()
- A. 归并排序 (Merge sort)
 - B. 快速排序 (Quick sort)
 - C. 克鲁斯卡尔 (Kruskal) 最小生成树算法
 - D. 快速傅里叶变换 (FFT, Fast Fourier Transform)
- 4) 设 n 是描述问题规模的非负整数, 下列程序的时间复杂度是多少? ()
- ```
int sum = 0;
for (int i = 0; i < n; ++i) {
 for (int j = 0; j < n; ++j) {
 sum += 1; // $\Theta(1)$
 }
}
```
- A.  $O(n^2)$
  - B.  $\Theta(n^2)$
  - C.  $\Omega(n^2)$
  - D.  $\Theta(n)$
- 5) 对于一个线性表既要求能够进行较快速地插入和删除, 又要求存储结构能反映数据之间的逻辑关系, 则应该选用 ( )。
- A. 顺序存储方式
  - B. 链式存储方式
  - C. 哈希表/散列表 (Hash table) 存储方式
  - D. 以上均可以
- 6) 函数  $T(n)$  表示当输入数据规模为  $n$  时的算法时间复杂度, 以下算法效率最优的是 ( )。
- A.  $T(n) = T(n-1) + 1, T(1) = 1$
  - B.  $T(n) = \sqrt{n}$
  - C.  $T(n) = \log_2(100n)$
  - D.  $T(n) = n^3$
- 7) 采用插入排序对下列四个序列进行递增排序, 比较次数最少的是? ( )
- A. 3,1,5,7,6,8
  - B. 7,1,8,6,3,5

- C. 3,1,6,8,5,7  
D. 1,3,5,7,6,8
- 8) 节点数量为  $n$  的完美二叉树 (Perfect binary tree) 高度是 ( )。注意: 此处定义二叉树的高度为  $h$ , 空树  $h=-1$ , 一个节点的树  $h=0$ , 两层树  $h=1$ ;  $\ln()$  为自然对数。  
A.  $\log_2(n+1)-1$     B.  $\log_2(n-1)+1$     C.  $\ln(n+1)-1$     D.  $\ln(n-1)+1$
- 9) 设高度为  $h$  的二叉树上只有度 (degree) 为 0 和度为 2 的结点, 则此类二叉树中所包含的结点数至少为 ( )。注意: 此处定义二叉树的高度为  $h$ , 空树  $h=-1$ , 一个节点的树  $h=0$ , 两层树  $h=1$ 。  
A.  $2h-1$     B.  $2h+1$     C.  $h+2$     D.  $h+1$
- 10) 具有  $n$  个顶点的连通无向图至少有 ( ) 条边。  
A.  $n$     B.  $n+1$     C.  $n-1$     D.  $2n$
- 11) 在最小堆 (Min-heap) 中查找最大元素的复杂度为 ( )。  
A.  $O(n)$     B.  $O(1)$     C.  $O(n\log_2(n))$     D.  $O(1.44\log_2(n))$
- 12) 在最大堆 (Max-heap) 中查找最大元素的复杂度为 ( )。  
A.  $O(n)$     B.  $O(1)$     C.  $O(n\log_2(n))$     D.  $O(1.44\log_2(n))$
- 13) 对于一个长度为  $m$  的循环队列, 其头指针  $front$  指向队列中第一个元素, 尾指针  $rear$  指向队列中最后一个元素, 则该队列为空的条件是 ( )。  
A.  $(rear+1) \% m = front$   
B.  $(rear+1) = front$   
C.  $(rear-1) = front$   
D.  $(rear-1) \% m = front$
- 14) 下面关于二叉树的叙述中, 正确的是 ( )  
A. 若有一个结点 (Node) 是二叉树中某个子树的中序遍历 (In-order traversal) 结果序列的最后一个结点, 则它一定是该子树的前序遍历 (Pre-order traversal) 结果序列的最后一个结点。  
B. 若有一个结点是二叉树中某个子树的前序遍历结果序列的最后一个结点, 则它一定是该子树的中序遍历结果序列的最后一个结点。  
C. 若有一个叶子结点 (Leaf node) 是二叉树中某个子树的中序遍历结果序列的最后一个结点, 则它一定是该子树的前序遍历结果序列的最后一个结点。  
D. 若有一个叶子结点是二叉树中某个子树的前序遍历结果序列的最后一个结点, 则它一定是该子树的中序遍历结果序列的最后一个结点。
- 15) 下面关于无向图的描述哪个是正确的 ( )  
A. 树是一个无环图  
B. 无环图一定是树  
C. 图的最小生成树上的一条路径是连接这条路径任意两个顶点的最短路径  
D. 以上均不对

### 3. 简答题 (8 题, 每题 5 分, 共 40 分)

请在答题纸上简要回答以下问题。

- 1) 请简述为什么完全二叉树的高度为  $\lceil \log_2(n) \rceil$ , 其中  $n$  为节点数量,  $\lceil \cdot \rceil$  是取整符号。
- 2) 使用最大堆 (Max-heap) 进行排序前, 需要初始化堆。写出输入序列 {5, 14, 17, 9, 12, 8, 19, 11, 16} 初始化后的最大堆。注意: 该堆以完全二叉树的数组形式存放。

|  |  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|--|

- 3) 请简述一个图是二分图 (Bipartite Graphs) 的充要条件。
- 4) 简述并查集 (Disjoint Set) 的主要应用场景。
- 5) 对于一个有序顺序表 (Sorted List) 来说, 二分查找是否在任何时候都比顺序查找快, 并简述理由?
- 6) 在堆排序 (Heap-sort)、快速排序和归并排序中:
  - i. 如果只从存储空间复杂度考虑, 则首选、次选和最后选择的排序方法分别是哪些? (2 分)
  - ii. 如果只从排序结果的稳定性考虑, 则应选择哪种排序方法? (1 分)
  - iii. 如果只从平均情况下时间复杂度考虑, 则应选取哪种排序方法? (1 分)
  - iv. 如果只从最坏情况下最快且要节省内存考虑, 则应选取哪种排序方法? (1 分)
- 7) 在一个只含有字符 a,b,c,d,e,f 的文本中, a,b,c,d,e,f 的使用频次分别是 23, 24, 25, 26, 27, 28, 根据这些统计信息, 可以针对该文本生成 Huffman (哈夫曼) 编码。请写出一组针对字符串 abcdef 的霍夫曼编码, 以及必要的推导过程。
- 8) 请写出判断单向链表 head 为空的判定条件。提示: 单向链表可分为带头结点的单向链表, 不带头结点的单链表, 带头结点的循环单向链表等。

### 4. 堆栈和队列 (3 题, 共 15 分)

请考虑下列一系列操作:

push(5), push(10), pop(), push(9), push(16), pop(), pop(), push(20)

请在答题纸上回答如下问题或者画出相应答案:

- 1) 假设上述操作发生在一个初始状态为空的队列 (Queue) 上, 请按顺序写出出队序列, 和操作结束后队列的最终状态。(4 分)
- 2) 假设上述操作发生在一个初始状态为空的堆栈 (Stack) 上, 请按顺序写出出栈序列, 和操作结束后堆栈的最终状态。(4 分)
- 3) 假设上述操作发生在一个初始状态如图 1 所示的最小二叉堆 (Binary Min-heap) 上, 请按顺序写出出堆序列和操作结束后最小堆的最终状态。(7 分)

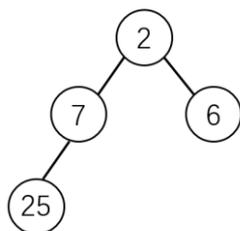


图 1-最小二叉堆 (Binary min-heap) 的初始状态

## 5. 大 O 表达排序 (4 题, 共 20 分)

请在答题纸上回答如下问题:

- 1) 请按照升序顺序对下列算法复杂度函数进行排序, 并将排序后函数对应的序号写在答题纸上。算法复杂度的升序定义为: 如果  $f(n)=O(g(n))$ , 那么  $f(n), g(n)$  是一个升序序列。(4 分)

1.  $n^3$ ; 2.  $n!$ ; 3.  $\log_2(2n)$ ; 4.  $2^{\sqrt{n}}$ ; 5.  $\sqrt{n}$ ; 6.  $n^n$ ; 7. 100;

- 2) 下列 5 段代码 ((a), (b), (c), (d), (e)) 描述了一个排序算法, 但 5 段代码的排序被打乱了, 请仔细阅读代码, 并写出算法实现的正确顺序。(5 分)

```
(a) List sort(List input){
 if(input.length <= 1) {return input.copy();}
```

```
(b) List sorted_left = left.sort();
 List sorted_right = right.sort();
 sorted_left.append(p);
 sorted_left.append(sorted_right);
```

```
(c) for (int i=0; i<input.length; i++){
 if (input[i] < p) {left.append(input[i]);}
 else {right.append(input[i]);}
}
```

```
(d) int p = input.poplast(); // Now, 'p' no longer exists in 'input'
 List left = new List();
 List right = new List(); // 'left' and 'right' are empty
```

```
(e) cout << sorted_left << endl;
 // print all elements in the array from left to right
 delete sorted_right;
 return sorted_left;}
```

- 3) 问题 2) 中所描述的算法与以下哪种排序算法最接近? (3 分)

- A. 冒泡排序  
B. 插入排序  
C. 归并排序  
D. 快速排序

- 4) 请使用问题 2) 中所描述的排序算法对数字序列 9,7,3,5,1,8,4,2,6 进行排序, 并写出算法完成后所有显示在电脑显示屏上的结果。(8 分)

**6. 二叉树 (8 题, 共 20 分)**

一个二叉搜索树 (Binary Search Tree) 的初始状态如图-2 所示:

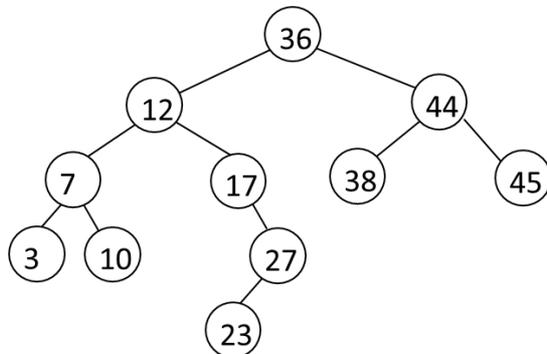


图 2-二叉搜索树的初始状态

请在答题纸上回答如下问题或者画出相应答案:

- 1) 请写出该树的后序遍历 (Post-order traversal) 结果。(2 分)
- 2) 请画出左子结点/右兄弟结点 (Left-child-right-sibling) 的方式所表示的树。(2 分)
- 3) 若 a 的 Next 节点为该树中第一个比 a 大的树的节点, 请分别写出 37, 11 的下一个 (Next) 节点。(2 分)
- 4) 不考虑树的平衡性, 请画出删除 36 节点后的二叉搜索树。(2 分)
- 5) 考虑树的 AVL 平衡特性, 即任意节点左右两边子树的高度差不大于 1, 在初始树中对 23 这个节点导致的非平衡性进行调整, 画出调整后的 AVL 树。(2 分)
- 6) 在问题 5) 获得的树的基础上, 同样考虑 AVL 平衡特性, 画出插入元素 2 后的树。(4 分)
- 7) 在 AVL 树中, 假设节点数量为 n, 那么可以证明整棵树的高度 h 在最坏情况下为  $[1.44 * \lg(n+1) - 1.33] \leq h \leq [1.44 * \lg(n+1) - 1.33] + 1$ , 请写出最好情况下树的高度, 即节点数量为 n 时 AVL 树的最小树高。(2 分)
- 8) 根据问题 7), 简要分析 AVL 树是否适合使用数组形式存储 (提示: 二叉树的数组存储类似二叉堆中在数组中的存储方式, 即节点的父子关系可以直接通过数组下标计算得出) (4 分)

**7. 图 (6 题, 共 15 分)**

给定如下带权无向图 (Weighted Undirected Graph) 如图-3 所示:

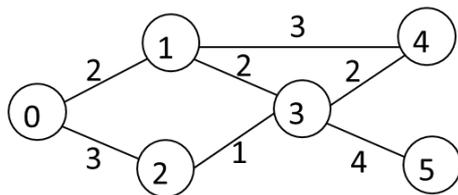


图 3-带权无向图

请在答题纸上回答如下问题或者画出相应答案:

- 1) 请写出从顶点 0 开始对该树的广度优先遍历结果。(注意：相邻节点中优先遍历序号小的节点) (2 分)
- 2) 求从顶点 0 出发的最小生成树 (Minimum Spanning Tree)，按普里姆 (Prim) 算法，写出逐次选择的顶点。(2 分)
- 3) 画出题 2) 中的最小生成树。(3 分)
- 4) 求从顶点 0 出发到各个顶点的最短路径，按迪杰斯特拉 (Dijkstra) 算法，写出逐次选择的顶点。(3 分)
- 5) 画出题 4) 中由顶点 0 到各个顶点的最短路径。(2 分)
- 6) 将图 3 中连接顶点  $i$  和顶点  $j$  的边记为  $(i, j)$ 。边  $(1, 4)$  在原始图中权值为 3，现对其权值进行改变，其它边的权值不变。假设边  $(1, 4)$  的权值为  $a$  ( $a > 0$  且为整数)，求使得顶点 0 到顶点 4 的最短路径通过边  $(3, 4)$  的  $a$  的取值范围。(3 分)