

华南理工大学
2011 年攻读硕士学位研究生入学考试试卷

(请在答题纸上做答, 试卷上做答无效, 试后本卷必须与答题纸一同交回)

科目名称: 数据结构

适用专业: 软件工程(专业学位)

本卷满分: 150 分

共 3 页

一、填空题(每空 2 分, 共 30 分)

1. 广义表 $A = (a, (b), (c, d))$, 则表头为_____, 表尾为_____, 长度为_____。

2. 设单链表的存储结构为:

```
typedef struct LNode {
    ElemType data;
    struct LNode *next;
} LNode, *LinkList;
```

若在 p 所指结点前插入 s 指向的新结点, 则应顺序执行语句_____和_____。

3. 给定数组 $A[m][n]$. 若 $A[0][0]$ 位于 $644_{(10)}$, $A[2][2]$ 存储在 $676_{(10)}$, 且每个元素占一个空间. “ $_{(10)}$ ”为十进制表示. 那么元素 $A[5][5]_{(10)}$ 的位置是_____。

4. 有一个占 4 字节的指针和一个需要 8 字节的数据元素, 当数组在少于_____满时链表实现比基于数组的实现要求的空间更少

5. 一个基于磁盘请求的程序运行时间最有效的节省方法是_____。

6. 若在线性表中采用折半查找法查找元素, 该线性表应满足条件_____。

7. 堆排序算法的平均情形的算法时间复杂度是_____。

8. 一个有 n 个内部节点的满三叉树, 它的总结点个数为_____。

9. 将具有 n 个结点的完全二叉树按层序编号, 则编号为 i 的结点, 其父亲节点编号为_____, 其右孩子编号 (假设该节点存在) 为_____。

10. 具有 n 个顶点的有向连通简单平面图最少有_____条边, 最多有_____条边。

二、判断题(20 分)

1. 快速排序是一种交换排序。

2. 在 $2n^2$, $30 \log n$, 5^n , $4n$ 中, 当 n 变大时所对应的增长率最有效率的算法是 $30 \log n$ 。

3. 链接存储方式不要求大块连续存储空间, 因此比顺序存储方式优越。

4. 外排序中，置换选择排序的功能是将顺序串进行合并。
5. 具有 n 个顶点和 $n-1$ 条边的无向图不一定是连通的。
6. 由二叉树的后序序列和中序序列可以唯一确定一棵二叉树。
7. 在选择排序方法中，关键字比较的次数与记录的初始排列次序无关。
8. 在一个 **BST** 中，任何节点的左孩子都比右孩子小，在一个堆中也是这样。
9. 完全树的叶结点都在层数最大的一层。
10. 在哈夫曼编码中，当两个字符出现的频率相同时，其编码也相同。

三、简答题（60分）

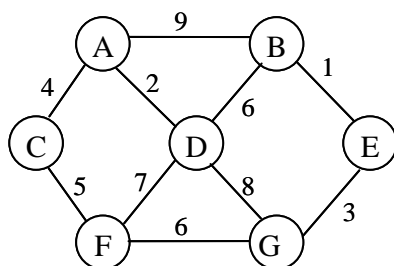
1. 给出关键字 34, 13, 5, 7, 90, 45, 4, 85, 33，请用起泡排序法对其升序排序，写出每趟结果，分析时间复杂度，并说明该排序算法是否为稳定的排序算法。

2. 假设字符 A,B,C,D,E,F,G,H 的使用频度如下表，请构造一棵 Huffman 树，并写出各字母的 Huffman（哈夫曼）编码，计算编码平均长度。

A	B	C	D	E	F	G	H
5	25	3	6	10	11	36	4

3. 使用封闭哈希 closed hashing 插入码 22, 41, 53, 46, 30, 13, 1, 67 到一个 11 槽的哈希表中(槽编号 0 到 10)，使用双哈希解决冲突，要使用的哈希函数为 H_1 和 H_2 定义为 $H_1(k) = 3k \bmod 11$ ， $H_2(k) = 7k \bmod 10 + 1$ 。画出所有 8 个码已被插入后的哈希表，描述如何使用 H_1 和 H_2 进行哈希。

4. 先用 Dijkstra 算法找出从 C 到所有其他节点的最短路径，再用 Kruskal's 算法找出最小生成树。



5. 根据一个元素类型为整型的单链表生成两个单链表，使得第一个单链表中包含原单链表中所有元素值为奇数的结点，使得第二个单链表中包含原单链表中所有元素值为偶数的结点，原单链表保持不变。

6. 设 n 个人围坐在一个圆桌周围, 现在从第 s 个人开始报数, 数到第 m 个人, 让他出局; 然后从出局的下一个人重新开始报数, 数到第 m 个人, 再让他出局, …… , 如此反复直到所有的人全部出局为止。下面要解决的 Josephus 问题是: 对于任意给定的 n, s 和 m , 求出这 n 个人的出局序列。请以 $n = 9, s = 1, m = 5$ 为例, 人工模拟 Josephus 的求解过程以求得问题的解。

四、计算题 (20 分)

1. 什么是静态索引结构? 什么是动态索引结构? 它们各有哪些优缺点?
2. 举出你所知的各种内排序算法, 根据算法复杂度对内排序算法进行分类, 并说明归类原因。

五、程序设计 (20 分)

1. 试设计一个实现下述要求的 *Locate* 运算的函数。设有一个带表头结点的双向链表 L , 每个结点有 4 个数据成员: 指向前驱结点的指针 *prior*、指向后继结点的指针 *next*、存放数据的成员 *data* 和访问频度 *freq*。所有结点的 *freq* 初始时都为 0。每当在链表上进行一次 *Locate* (L, x) 操作时, 令元素值为 x 的结点的访问频度 *freq* 加 1, 并将该结点前移, 链接到与它的访问频度相等的结点后面, 使得链表中所有结点保持按访问频度递减的顺序排列, 以使频繁访问的结点总是靠近表头。
2. 设有一个背包可以放入的物品的重量为 s , 现有 n 件物品, 重量分别为 $w[1], w[2], \dots, w[n]$ 。问能否从这 n 件物品中选择若干件放入此背包中, 使得放入的重量之和正好为 s 。如果存在一种符合上述要求的选择, 则称此背包问题有解(或称其解为真); 否则称此背包问题无解(或称其解为假)。试用递归方法设计求解背包问题的算法。(提示: 此背包问题的递归定义如下:)

$$KNAP(s, n) = \begin{cases} \text{True} & s = 0 & \text{此时背包问题一定有解} \\ \text{False} & s < 0 & \text{总重量不能为负数} \\ \text{False} & s > 0 \text{ 且 } n < 1 & \text{物品件数不能为负数} \\ KNAP(s, n-1) \text{ 或} & s > 0 \text{ 且 } n \geq 1 & \text{所选物品中不包括 } w[n] \text{ 时} \\ KNAP(s - w[n], n-1) & & \text{所选物品中包括 } w[n] \text{ 时} \end{cases}$$