

# 数据结构期末试卷答案 2005/01/08

班级\_\_\_\_\_姓名\_\_\_\_\_学号\_\_\_\_\_成绩\_\_\_\_\_

## 一、填空和选择(共 15 分)

1. 在  $n$  个元素的顺序表中插入或删除一个元素, 需平均移动  $n/2$  个元素。
2. 在双向链表  $p$  所指结点之后插入  $s$  所指结点的操作是 **D**  
A.  $p \rightarrow \text{right} = s; s \rightarrow \text{left} = p; p \rightarrow \text{right} \rightarrow \text{left} = s; s \rightarrow \text{right} = p \rightarrow \text{right};$   
B.  $p \rightarrow \text{right} = s; p \rightarrow \text{right} \rightarrow \text{left} = s; s \rightarrow \text{left} = p; s \rightarrow \text{right} = p \rightarrow \text{right};$   
C.  $s \rightarrow \text{left} = p; s \rightarrow \text{right} = p \rightarrow \text{right}; p \rightarrow \text{right} = s; p \rightarrow \text{right} \rightarrow \text{left} = s;$   
D.  $s \rightarrow \text{left} = p; s \rightarrow \text{right} = p \rightarrow \text{right}; p \rightarrow \text{right} \rightarrow \text{left} = s; p \rightarrow \text{right} = s;$
3. 在数据结构中, 从逻辑上可以把数据结构分成 **C**。  
A. 动态结构和静态结构    B. 紧凑结构和非紧凑结构  
C. 线性结构和非线性结构    D. 内部结构和外部结构
4. 算法分析的两个主要方面是 **A**。  
A. 空间复杂性和时间复杂性    B. 正确性和简明性  
C. 可读性和文档性    D. 数据复杂性和程序复杂性
5. 线性表的顺序存储结构是一种 **A** 的存储结构, 线性表的链式存储结构是一种 **B** 的存储结构。  
A. 随机存取    B. 顺序存取    C. 索引存取    D. 散列存取
6. 向量、栈和队列都是 线性 结构, 可以在向量的 任意 位置插入和删除元素; 对于栈只能在 栈顶 插入和删除元素; 对于队列只能在 队尾 插入元素和在 队头 删除元素。
7. 设循环队列中数组的下标范围是  $1 \sim n$ , 其头尾指针分别为  $f$  和  $r$ , 则其元素个数为 **D**。  
A.  $r-f$     B.  $r-f+1$     C.  $(r-f) \bmod n + 1$     D.  $(r-f+n) \bmod n$
8. 将下三角矩阵  $A[7,7]$  的下三角部分逐行地存储到起始地址为 1000 的内存单元中(下标从 0 开始, 不存储上三角部分), 已知每个元素占 4 个单元, 则  $A[5,4]$  的地址是 **1076**。
9. 某二叉树的前序遍历结点访问顺序是  $abdgcefh$ , 中序遍历的结点访问顺序是  $dgbaechf$ , 则其后序遍历的结点访问顺序是 **D**  
A.  $bdgcefh$     B.  $gdbecfha$     C.  $bdgaeCHF$     D.  $gdbehfca$

## 二、简答题

1. (4 分)求一棵满  $k$  叉树上的叶子结点数  $n_0$  和非叶子结点数  $n_k$  之间满足的关系。

总结点数:  $n_0 + n_k$                   总分支数:  $k * n_k$

所以:  $n_0 + n_k = k * n_k + 1$

$$n_0 = (k-1) n_k + 1$$

2. (4 分)假设高度为  $H$  的二叉树上只有度为 0 和度为 2 的结点, 问此类二叉树中结点数可能达到的最大值和最小值各是多少?

最大值:  $2^H - 1$

最小值:  $2H - 1$

3. (6 分)根据下面的矩阵, 写出相应的三元组表, 并写出矩阵转置后的三元组表。(不要求过程)

$$\begin{pmatrix} 0 & 12 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 & 0 \\ -3 & 0 & 0 & 0 & 0 & 14 & 0 \\ 0 & 0 & 13 & 0 & 0 & 0 & 0 \\ 0 & 18 & 0 & 0 & 0 & 0 & 0 \\ 15 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

三、(10 分) 判断以下序列是否为小(顶)根堆? 若否, 则以最少的移动次数将它们调整为小(顶)根堆。(要求画出最后的堆结构和线性序列)

(1) (19, 78, 32, 66, 26, 58, 46, 95, 89, 31)

(2) (113, 98, 69, 35, 68, 25, 43, 19, 31, 55, 16, 29)

(1) 否(19 26 32 66 31 58 46 95 89 78)

(2) 否(16 19 25 31 55 29 43 35 113 98 68 69)

四、(8 分) 设有关键码序列(Q, H, C, Y, Q, A, M, S, R, D, F, X), 要求按照关键码值递增的次序进行排序。

(1) 若采用初始步长为 4 的 Shell(希尔)排序法, 写出一趟排序的结果;

(2) 若采用以第一个元素为分界元素(枢轴)的快速排序法, 写出一趟排序的结果。

(1) Q A C S Q D F X R H M Y

(2) F H C D Q A M Q R S Y X

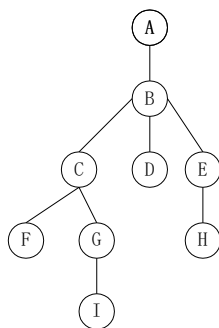
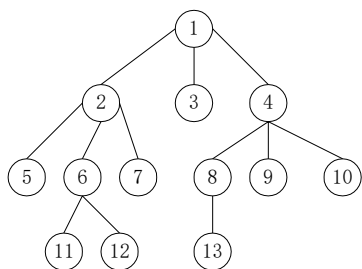
## 六、综合题

1. 以下各图:

(1) (4 分)分别画出和下列树对应的各个二叉树;

(2) (4 分)对各棵树按树的遍历规则给出先根遍历序列、后根遍历序列;

(3) (4 分)对(1)中得到的各棵二叉树给出先根遍历序列、后根遍历序列。



(2)先根: 1 2 5 6 11 12 7 3 4 8 13 9 10

后根: 5 11 12 6 7 2 3 13 8 9 10 4 1

先根: ABCFGIDEH

后根: FJGCDHEBA

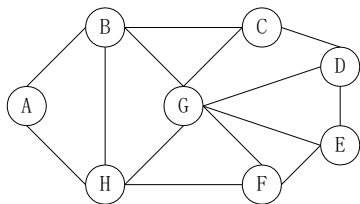
(3)先根: 1 2 5 6 11 12 7 3 4 8 13 9 10

后根: 12 11 7 6 5 13 10 9 8 4 3 2 1

先根: ABCFGIDEH

后根: IGFHEDCBA

2. (10 分) 画出下图所示的无向图的邻接表 (顶点由 A 到 H 排列), 并根据所得邻接表给出深度优先和广度优先搜索遍历该图所得的顶点序列。



深度优先: ABCDEFGH

广度优先: ABHCGFDE

3. (11 分) 假设用于通信的电文仅由 10 个字符 (A, B, C, D, E, F, G, H, I, J) 组成, 字符在电文中出现的频率分别为 (0.10, 0.19, 0.02, 0.05, 0.17, 0.03, 0.21, 0.07, 0.15, 0.01)。

(1) 画出哈夫曼树;

(2) 最长的编码为几位, 对应哪些字符? 最短的编码为几位, 对应哪些字符。

(3) 计算其带权路径长度。

(2) 6 C J      2   B G

(3)  $[6*(1+2)+5*3+4*(5+7+10)+3*(15+17)+2*(19+21)]/10=29.7$

4. (6 分) 设一个散列表包含  $\text{hashSize}=13$  个表项, 其下标从 0 到 12, 采用链地址(拉链)法解决冲突. 请按以下要求, 将下列关键码散列到表中.

(10   100   32   45   58   126   3   29   200   400   0)

(1) 散列函数采用除留余数法, 用  $\% \text{hashSize}$  (取余运算) 构造 hash 表;

(2) 并计算查找成功时的平均查找长度。

(2)  $\text{ASL}=(1*6+2*4+3*1)/11 = 1.55$

七、(14 分)完成下列折半(二分)查找算法  
(ST 为顺序表，key 为要查找的关键字。)

```
.  
int Search_Bin(SSTable ST, KeyType key){ //  
    low=1;    high=ST.length;  
    while ( __low<=high__ ){  
        mid=__ (low+high)/2 __;  
        if ( __key ==ST[mid].key__ )  
            return __ mid __;  
        else if ( __key<ST[mid].key__ )  
            high=__ mid-1 __;  
        else low=__ mid+1 __;  
    }  
    return 0;  
}
```