

杭州电子科技大学

2007 年攻读硕士学位研究生入学考试

《数据结构》试题

(试题共六大题, 5 页, 150 分)

姓名_____报考专业_____准考证号_____

【所有答案必须写在答题纸上, 做在试卷或草稿纸上无效!】

一、判断题 (每小题 2 分, 共 20 分)

判断以下各题是否正确, 若正确答 T, 否则答 F。

1. 数据元素是数据的基本单位。
2. 对于具有 n 个结点的二叉树, 不论其形态如何, 进行先序、中序或后序遍历的时间复杂度均为 $O(n)$ 。
3. 为度量一个查找算法的性能, 需要在时间和空间方面进行权衡。
4. 一棵 m 阶 B 树中每个结点最多有 m 棵子树, 最少有 2 棵子树。
5. 在赫夫曼编码中, 当两个字符出现的频率相同时, 其编码也相同, 对于这种情况应当特殊处理。
6. 串在用定长顺序存储表示时, 如果实际串值长度超过预定义长度, 则用压缩方式存储。
7. 用邻接矩阵法存储一个无向图时, 所需的存储空间大小与图的边数无关。
8. 广义表的表头和表尾都有可能是原子或广义表。
9. 数据的逻辑结构和物理结构密切相关, 一个算法的设计取决于数据的逻辑结构, 而其实现依赖于数据的存储结构。
10. 最差拟合法是动态存储分配的一种方法, 该方法将可利用空间表中最大的空闲块 (若该块的大小不小于用户请求分配的存储量) 的一部分分配给用户。

二、选择题 (每小题 2 分, 共 20 分)

从供选择的答案中选择一个最合适的答案。

1. 线性表的链式存储结构是一种()的存储结构。
A. 散列存取 B. 索引存取 C. 顺序存取 D. 随机存取
2. 在一个含有 n 个元素的有序表上进行折半查找, 找到一个元素最多要进行()次元素比较。
A. $\lfloor \log_2(n) \rfloor$ B. $\lfloor \log_2(n) \rfloor + 1$
C. $\lfloor \log_2(n+1) \rfloor$ D. $\lfloor \log_2(n+1) \rfloor + 1$
3. 用三叉链表作二叉树的存储结构, 当二叉树中有 n 个结点时, 有()个空指针。
A. $n-1$ B. n C. $n+1$ D. $n+2$

4. 用链表表示线性表的优点是 ()。
- A. 便于随机存取 B. 花费的存储空间比顺序表少
- C. 便于插入与删除 D. 数据元素的物理顺序与逻辑顺序相同

5. 一个二叉树按顺序方式存储在一个维数组中，如下所示：

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	B	C		D	E	F					G		H	I

则结点 H 在二叉树的第 () 层。(根结点为第一层)

- A. 1 B. 2 C. 3 D. 4
6. 若需要从 100 万个记录中挑选出关键字最小的 50 个记录，下列方法中最好使用 () 方法。
- A. 插入排序 B. 希尔排序 C. 快速排序 D. 堆排序
7. 若顺序栈 S 的类型为：

```
typedef struct
{
    SElemType *base;
    int top; //top 为栈顶指示器，初始化时，将 top 置为-1
    int stacksize;
} SqStack;
```

则向 S 中压入新元素时，应当 ()。

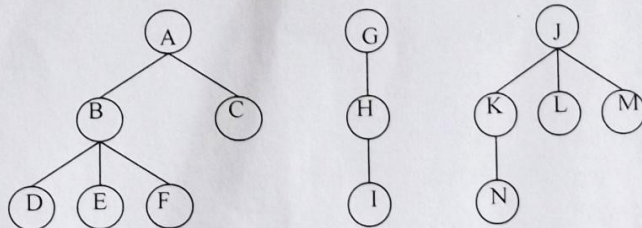
- A. 先移动栈顶指示器，再在栈顶指示器所指处存入元素
- B. 先在栈顶指示器所指处存入元素，再移动栈顶指示器
- C. 先后次序无关紧要，只要在栈顶指示器所指处存入元素即可
- D. 同时进行
8. 下列查找方法中，() 适用于查找单链表。
- A. 顺序查找 B. 二分查找 C. 分块查找 D. 哈希查找
9. 在一个有向图中，所有顶点的出度之和等于所有弧数的 () 倍。
- A. 3 B. 2 C. 1 D. 1/2
10. 对 5 个不同的数据元素进行直接插入排序，最少需要进行 () 次比较。
- A. 4 B. 5 C. 8 D. 10

三、填空题 (每小题 2 分，共 10 分)

1. 结点数为 70 的二义排序树高度至少为_____。
2. 已知广义表 $L = (((x), y, ((z))), ((a, b, c)), d)$ ，利用 head (T) (取 T 的表头) 和 tail (T) (取 T 的表尾) 两个运算，则从 L 表中取出原子项 y 的运算是_____。
3. 不带头结点的单链表 head 为空的判断条件是_____。
4. 当二叉排序树是一棵平衡树时，其平均查找长度为_____。
5. 有 m 个叶子结点的赫夫曼树，其结点总数是_____。

四、应用题（每小题 8 分，共 64 分）

1. 已知一棵二叉树的后序序列和中序序列分别为：C, B, G, H, F, E, D, A 和 B, C, A, D, G, F, H, E，试画出该二叉树，并将其中序线索化。
2. 设哈希表的长度为 15，哈希函数为 $H(k) = k \% 13$ ，给定的关键字序列为 19, 14, 23, 01, 68, 20, 80, 27。试画出用二次探查法处理冲突时所构成的哈希表；并求在等概率情况下查找成功时的平均查找长度。
3. 对关键字序列 (51, 28, 9, 36, 46, 78, 48, 4, 20)，构造一棵 2-3 树，画出每加入一个新结点后的结果。
4. 将如下所示森林转换成二叉树，并分别写出该森林的先序序列和中序序列。



5. 对关键字序列 37, 72, 20, 46, 65, 54, 52, 18, 26, 30，分别采用快速排序、基数排序和堆排序方法，由小到大进行排序，请写出
 - (1) 快速排序第一趟排序的结果；
 - (2) 基数排序第一趟排序的结果；
 - (3) 堆排序将初始序列建成堆的结果。
6. 若有带权的无向图 $G = (V, E)$ ，其中顶点集 $V = \{a, b, c, d, e, f\}$ ，边集 $E = \{(a, b), (a, c), (b, c), (b, d), (c, e), (c, f), (d, e), (e, f)\}$ ，各边对应的权值依次为 (4, 10, 8, 5, 6, 7, 12, 9)。
 - (1) 写出其邻接矩阵；
 - (2) 用 Prim 算法，从顶点 a 开始构造最小生成树，依次列出加入生成树的边。
7. 设在某通信系统中仅使用十个符号，它们的使用频率分别为 0.04, 0.15, 0.09, 0.03, 0.17, 0.10, 0.14, 0.11, 0.12, 0.05。试构造一棵赫夫曼树并为这十个符号设计赫夫曼编码。
8. 已知广义表 $L = ((x, y, ((z))), ((a, b, c)), d)$ ，请给出广义表的结点结构定义（形式定义或图示均可），并根据你定义的结点结构，画出该广义表的存储结构图。

五、阅读理解题（每小题 6 分，共 12 分）

1. 阅读下列算法，并补充所缺条件/语句。

```

结点类型为： typedef struct LNode
                { ElemType data;
                  struct LNode *next;
                } LNode, *LinkList;

void delete_same (LinkList la)
{ // 从头指针为 la 的带头结点的有序单链表中删除所有值

```



```

//相同的多余元素，并释放被删结点空间。
p = la->next;
while (p)
{
    q = p->next;
    while (_____)
    {
        p->next=q->next;
        free (q);
        q=p->next;
    }
    _____;
}
}

```

2. 已知二叉树中的结点类型定义为：

```

typedef struct BiTreeNode
{
    ElemType data; //结点元素值域
    BiTreeNode *lchild, *rchild;
    // lchild 和 rchild 分别为指向左、右孩子结点的指针域
} BiTreeNode, BiTree;

```

根据下面函数的定义指出函数的功能。

```

BiTreeNode * BinTreeS ( BiTreeNode * BT )
{
    // BT 指向二叉树的根结点。
    if ( !BT ) return NULL;
    else {
        pt = (BiTree) malloc(sizeof (BiTreeNode));
        pt->data = BT->data;
        pt->rchild = BinTreeS ( BT->lchild );
        pt->lchild = BinTreeS ( BT->rchild );
        return pt;
    }
}

```

六、算法设计题（本题共 24 分）

1. 设有一个带头结点的、按元素值递减有序排列的双向循环链表，编写算法，插入一个元素并保持其有序性。（本小题 10 分）

双向链表结点的类型定义如下：

```

typedef struct DuLNode
{
    int data;
    struct DuLNode *prior; //指向前驱的指针
    struct DuLNode *next; //指向后继的指针
} DuLNode, *DuLinkList;

```

2. 设有向图的十字链表存储结构定义如下，试分别编写求图中顶点 i 的出度和入度的算法。（本小题 14 分）

```
#define MAX_V_NUM 20
typedef struct ArcNode //弧结点
{ int tailvex, headvex; //该弧的尾和头顶点的位置
  struct ArcNode *hlink, *tlink; //分别为弧头相同和弧尾相同的弧的链域
} ArcNode;
typedef struct VNode
{ VertexType data; // 顶点信息
  ArcNode *firstin, *firstout; // 分别指向该顶点的第一条入弧和出弧
} VNode;
typedef struct
{ VNode xList[MAX_V_NUM];
  int vexnum, arcnum; //有向图的当前顶点数和弧数
} OLGraph;
```