

武汉科技学院

2008 年招收硕士学位研究生试卷

科目代码 806	科目名称 数据结构 (A)
考试时间 2008 年 1 月 20 日下午	报考专业 计算机应用技术

- 1、试题内容不得超过画线范围，试题必须打印，图表清晰，标注准确。
- 2、试题之间不留空格。
- 3、答案请写在答题纸上，在此试卷上答题无效。

题号	一	二	三	四	五	六	七	八	九	十	十一	得分
得分												

本试卷总分 150 分，考试时间 3 小时。

一、单项选择题（每题 2 分，共 40 分）

- 1、下列数据结构中，_____是非线性结构。
A. 栈 B. 队列 C. 完全二叉树 D. 堆
- 2、下面程序段的时间复杂度为_____。

```

for(i=0; i<n; i++)
    for(j=0; j<n; j++)
        { ++x; s += x; }

```

A. $O(1)$ B. $O(n)$ C. $O(n^2)$ D. $O(n^3)$
- 3、带头结点的单链表 head 不为空链表的判断条件是_____。
A. head == NULL B. head != NULL
C. head->next == NULL D. head->next != NULL
- 4、在单链表中，指针 p 指向结点 A，若要在 A 结点之后插入指针 s 指向的新结点，则指针的操作方式为_____。
A. p->next = s; s->next = p->next; B. s->next = p->next; p->next = s;
C. p->next = s; s = p->next; D. s->next = p->next; p->next = s->next;

- 5、在双向循环链表中，在 p 所指的结点之后插入指针 s 所指的新结点，其操作步骤是_____。
- A. $p \rightarrow next = s; s \rightarrow prior = p; p \rightarrow next \rightarrow prior = s; s \rightarrow next = p \rightarrow next;$
 B. $p \rightarrow next = s; p \rightarrow next \rightarrow prior = s; s \rightarrow prior = p; s \rightarrow next = p \rightarrow next;$
 C. $s \rightarrow prior = p; s \rightarrow next = p \rightarrow next; p \rightarrow next = s; p \rightarrow next \rightarrow prior = s;$
 D. $s \rightarrow prior = p; s \rightarrow next = p \rightarrow next; p \rightarrow next \rightarrow prior = s; p \rightarrow next = s;$
- 6、设 C 语言数组 Data[m] 作为循环队列的存储空间，front 为队头指针，rear 为队尾指针，则求队列长度的表达式是_____。
- A. $(rear - front) \% m;$ B. $(rear - front + m) \% m;$
 C. $(front - rear) \% m;$ D. $(front - rear + m) \% m;$
- 7、设有一顺序栈 T，元素 s1, s2, s3, s4, s5, s6 依次进栈，如果 6 个元素出栈的顺序是 s2, s3, s4, s6, s5, s1，则栈的容量至少应该是_____。
- A. 2 B. 3 C. 5 D. 6
- 8、设有一个字符串 S = "windows"，其子串的数目是_____个。
- A. 26 B. 27 C. 28 D. 29
- 9、设 S = "wuse" 则 StrLength 为求串长的函数，则 StrLength(S) = _____。
- A. 4 B. 5 C. 6 D. 7
- 10、在一棵度为 3 的树中，度为 3 的结点个数为 2，度为 2 的结点个数为 1，则度为 0 的结点个数为_____。
- A. 4 B. 5 C. 6 D. 7
- 11、已知树中结点总数为 n，则树中所有结点的度之和为_____。
- A. n B. n-1
 C. 2*n D. n*(n-1)
- 12、二叉树从 1 开始进行连续编号，要求每个结点的编号大于其左右孩子的编号，同一个结点的左右孩子中，其左孩子的编号小于其右孩子的编号，则可采用_____遍历方式实现编号。
- A. 先序遍历 B. 中序遍历 C. 后序遍历 D. 层次遍历

- 13、设有向图的顶点个数为 n ，则该图最多可以有_____条弧。
 A. $n - 1$ B. $n(n - 1)/2$ C. $n(n - 1)$ D. n^2
- 14、一个 n 个顶点的连通无向图，其边的个数至少为_____。
 A. $n - 1$ B. n C. $n + 1$ D. $n*(n - 1)$
- 15、已知有向图 $G=(V,E)$ ，其中 $V=\{V_1,V_2,V_3,V_4,V_5,V_6,V_7\}$ ，
 $E=\{<V_1,V_2>,<V_1,V_3>,<V_1,V_4>,<V_2,V_5>,<V_3,V_5>,<V_3,V_6>,<V_4,V_6>,<V_5,V_7>,<V_6,V_7>\}$ ， G 的拓扑序列是_____。
 A. $V_1,V_3,V_4,V_6,V_2,V_5,V_7$ B. $V_1,V_3,V_2,V_6,V_4,V_5,V_7$
 C. $V_1,V_3,V_4,V_5,V_2,V_6,V_7$ D. $V_1,V_2,V_5,V_3,V_4,V_6,V_7$
- 16、对于具有 256 个记录的文件，若采用分块查找法，且每块长度为 16，则平均查找长度为_____。
 A. 16 B. 17 C. 18 D. 不确定
- 17、设散列表长 $m=14$ ，哈希函数 $H(\text{key})=\text{key}\% 11$ 。表中已有 4 个结点。 $\text{Addr}(15)=4$ ； $\text{addr}(38)=5$ ； $\text{addr}(61)=6$ ； $\text{addr}(84)=7$ ，其余地址为空。如果用二次探查法处理冲突，关键字为 49 的结点的地址是_____。
 A. 11 B. 9 C. 8 D. 6
- 18、下列四个序列中，哪一个是堆_____。
 A. 16, 72, 31, 23, 94, 53 B. 94, 23, 31, 72, 16, 53
 C. 16, 53, 23, 94, 31, 72 D. 16, 23, 53, 31, 94, 72
- 19、在下列的排序方法中，_____方法可能出现这种情况：在最后一趟开始之前，所有的元素都不在其最终应在的正确位置上。
 A. 快速排序 B. 冒泡排序 C. 堆排序 D. 插入排序
- 20、下面四种内排序方法中，需要辅助空间最大的是_____。
 A. 快速排序 B. 堆排序 C. 归并排序 D. 希尔排序

二、判断题（每题 1 分，共 10 分）

- 1、在单链表中，要取得某个元素，只要知道该元素的指针即可，因此，单链表是随机存取的存储结构。

- 2、顺序存储方式只能用于存储线性结构。
- 3、设一个栈的输入序列为 A、B、C、D，则借助一个栈所得到的输出序列不可能是 A、C、D、B。
- 4、空串是任意串的子串。
- 5、将一棵树转换成二叉树（即，孩子兄弟树）存储，则根结点一定没有左子树。
- 6、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。
- 7、关键路径是 AOE 网（以边表示活动的网）中，从源点到终点的最长路径。
- 8、二叉排序树的查找和折半查找的时间性能相同。
- 9、基数排序是将多关键字分配排序的高位优先排序法用于单关键字的排序方法。
- 10、堆排序是稳定的排序方法。

三、填空题（每题 2 分，共 20 分）

- 1、数据结构的形式化定义为 (D, S) ，其中 D 是_____的有限集， S 是 D 上关系的有限集。
- 2、数据元素之间的关系在计算机中有两种不同的表示方法：顺序映象和非顺序映象，由此得到两种不同的存储结构是顺序存储结构和_____存储结构。
- 3、一个算法具有 5 个特性：_____、确定性、可行性、有零个或多个输入、一个或多个输出。
- 4、长度为 n 的顺序表，在任何位置上删除一个元素的概率相等，删除一个元素时平均移动_____个元素。
- 5、将长度为 n 的单链表链接在长度为 m 的单链表之后的算法的时间复杂度为_____。
- 6、最大容量为 m 的循环队列，队尾指针是 $rear$ ，队头指针是 $front$ ，则判断队列为空的条件是：_____。
- 7、设两个字符串分别为： $S1="Today is "$ ， $S2="20 Jan, 2008."$ ，调用联接函数 $Concat(T, S1, S2)$ 后，串 T 为：_____。
- 8、用二叉链表表示 n 个结点的二叉树时，必定存在_____个空链域。
- 9、从平均性能而言，_____排序是排序算法中最快的一种。

10、Prim（普里姆）算法的时间复杂度是 $O(n^2)$ ，它适用于求_____网的最小生成树；Kruskal（克鲁斯卡尔）算法的时间复杂度是 $O(e \log e)$ ，它适用于求_____网的最小生成树。

四、算法填空题（每空 2 分，共 20 分）

1、在顺序表 L 的第 i 个元素之前插入新的元素 e。

```
#define OK 1
#define ERROR 0
#define OVERFLOW -2
#define LIST_INIT_SIZE 100 // 表存储空间的初始分配量
#define LISTINCREMENT 10 // 表存储空间的分配增量
typedef struct {
    ElemType *elem; // 存储空间基址
    int length; // 当前长度
    int listsize; // 存储容量
} SqList;
typedef int Status;
Status ListInsert_Sq(SqList &L, int i, ElemType e)
{ if (i < 1 || i > L.length+1) return ERROR; // 插入位置不合法
  if (____(1)____) // 当前存储空间已满，增加分配
  { newbase=(ElemType*)realloc(L.elem,
    (L.listsize+LISTINCREMENT)*sizeof(ElemType));
    if (!newbase) exit(OVERFLOW); // 存储分配失败
    L.elem = newbase; // 新基址
    L.listsize += LISTINCREMENT; // 增加存储容量
  }
  q = &(L.elem[i-1]); // q 指示插入位置
```

```

    for (p = &(L.elem[L.length-1]); p >= q; -p)
        _____(2)_____; //插入位置及之后的元素后移
    *q=e; //插入 e
    ++L.length; //表长增一
    return OK;
} // ListInsert_Sq

```

2、在带头结点的单链表 L 中，删除第 i 个元素，并由 e 返回其值。

```

#define OK 1
#define ERROR 0
typedef struct LNode{
    ElemType data;
    struct LNode *next;
}LNode, *LinkList;
typedef int Status;
Status ListDelete_(LinkList &L, int i, ElemType &e)
{
    p = L; j = 0;
    while (p->next && j < i - 1) // 寻找第 i 个结点，并令 p 指向其前趋
    {
        _____(3)_____; ++j; //p 指针指向下一个元素
    }
    if (!(p->next) || j > i - 1) // 删除位置不合理
        return ERROR;
    q = p->next; p->next = q->next; // 删除结点
    e = q->data; _____(4)_____; //释放已经删除的结点
    return OK;
} // ListDelete_L

```

3、 reara、 rearb 是循环链表 A、 B 的尾指针， 将 B 合并到 A 的尾部。

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

LinkList Connect(LinkList reara , LinkList rearb)
{
    p=reara ->next;           //p 为表 A 的头指针
    reara->next=__(5)__;      //将表 B 连接到表 A 的尾
    free(rearb->next);        //释放表 B 的头结点
    rearb->next=p;           //将表 B 的尾结点的指针域指向表 A 的头
    return __(6)__;         //返回新循环链表的尾指针
}
```

4、 计算二叉树的叶子结点个数。

```
typedef int TElemType;
typedef struct BiTNode {
    TElemType data;
    struct BiTNode *lchild, *rchild; // 左右孩子指针
} BiTNode, *BiTree;

int Leaf(BiTree T)
{
    if(T)
    {
        if(__(7)__)           //如果 T 是叶子结点
            return 1;
        else return Leaf(T->lchild)+__(8)__; //左右子树叶子结点个数之和
    }
    else return 0;
}
```

5、有序表的查找

```
typedef int KeyType;
typedef struct
{   KeyType key;
}ElemType;
typedef struct
{   ElemType *elem;
    int length;
}SSTable;
int Search_Bin(SSTable ST, KeyType key)
{   low=1;   high=ST.length
    while(low <= high) {
        mid = _____ (9) _____;
        if EQ(key, ST.elem[mid].key)           //找到待查元素
            return mid;
        else if LT(key, ST.elem[mid].key) //继续在前半区间查找
            high = mid - 1;
        else _____ (10) _____; }           //继续在后半区间查找
    return 0;
}
```

五、计算与证明题（每题 4 分，共 20 分）

1、计算下面序段的时间复杂度。

```
for (i=1; i<=n; ++i)
    for (j=1; j<=i; ++j)
        for (k=1; k<=j; ++k)
            {
                x++;
            }
```

2、假设 1 个字符占 8 位，1 个指针占 32 位。已知串的块链存储表示如下：

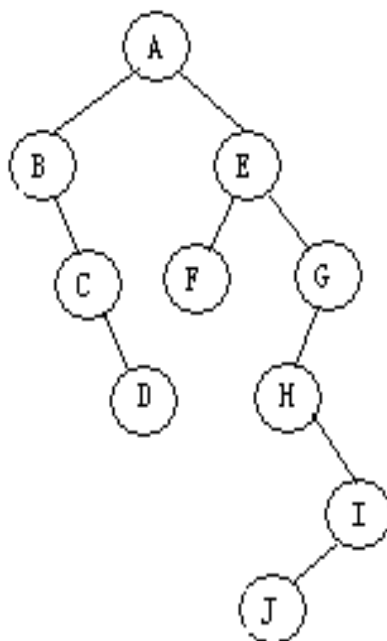
```
#define CHUNKSIZE 36 // 一块中字符的个数
typedef struct Chunk { // 结点结构
    char ch[CHUNKSIZE];
    struct Chunk *next;
} Chunk;
```

计算此块链存储结构的存储密度。

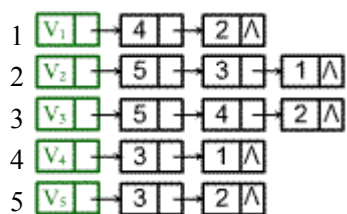
- 以 {5, 7, 8, 14, 3, 11} 作为叶子结点的权值构造赫夫曼树，计算此树的带权路径长度。
- 已知序列 {45, 24, 53, 37, 93}，建立一个二叉排序树。计算在等概率情况下查找成功的平均查找长度。
- 证明对任何一棵二叉树 T，如果其终端结点的个数为 n_0 ，度为 2 的结点个数为 n_2 ，则 $n_0 = n_2 + 1$ 。

六、解答题（每题 6 分，共 30 分）

1、将下图所示的二叉树转化为森林。



- 2、已知一棵二叉树的按层遍历序列是 ABCDEFGHIJ，中序遍历序列是 DBGEAHFIJC，①构造出该二叉树；②给出该二叉树的先序遍历序列；③给出该二叉树的后序遍历序列。
- 3、已知无向图的邻接表如下，①画出这个图；②写出该图的邻接矩阵 A；③根据邻接表，写出从顶点 V_1 出发，深度优先遍历该图所得到的顶点序列。



提示：

$$A[i][j] = \begin{cases} 1 & \text{若}(v_i, v_j)\text{或}\langle v_i, v_j \rangle\text{是}E(G)\text{中的边} \\ 0 & \text{若}(v_i, v_j)\text{或}\langle v_i, v_j \rangle\text{不是}E(G)\text{中的边} \end{cases}$$

- 4、已知一组关键字为 (19, 5, 21, 24, 45, 20, 68, 27, 70, 11, 10)，表长 15，请按哈希函数 $H(\text{key}) = \text{key} \text{ MOD } 13$ 和线性探测处理冲突。①构造该哈希表；②说明查找关键字 10 必须依次和表中哪些关键字比较。
- 5、给出一组关键字序列 {49, 38, 65, 97, 76, 13, 27, 49}，写出用下列算法按从小到大排序，第一趟结束时的序列。
- ① 希尔排序 (第一趟排序的增量为 5)；
 - ② 快速排序 (选第一个记录为枢轴(分隔))；
 - ③ 链式基数排序 (基数为 10)。

七、算法设计题 (共 10 分)

试写一个算法，删除带头结点单链表 L 中所有大于 min 且小于 max 的元素 (若表中存在这样的元素) 同时释放被删除结点的空间 (min 和 max 是给定的两个参变量，它们的值可以和表中的元素相同，也可以不同)。

```

typedef int ElemType;
typedef struct LNode
{
    ElemType data;
    struct LNode *next;
}LNode, *LinkedList;

void Delete_Between(LinkedList &L, ElemType min, ElemType max) // 函数首部
  
```

