

华北水利水电学院 2007 年攻读硕士学位研究生招生命题考试

数据结构 试题

注意事项：1、答案全部答在答题纸上，写在试卷上无效；

2、考试时间 180 分钟（3 个小时），满分 150 分。

3、算法题要求：（1）可以先用文字描述算法的解题思路；（2）用类 C 或类 Pascal 写出算法；（3）算法中应给出适当的注解。

一. 选择题（每小题 2 分，本题共 30 分）

- 在数据结构中，从逻辑上可以把数据结构分成_____。
A、动态结构和静态结构 B、紧凑结构和非紧凑结构
C、线形结构和非线形结构 D、内部结构和外部结构
- 队列与一般的线性表的区别在于_____。
A、数据元素的类型不同
B、运算是否受限制
C、数据元素的个数不同
D、逻辑结构不同
- 若已知一个栈的入栈序列是 1, 2, 3, ..., n, 其输出序列是 p₁, p₂, p₃, ..., p_n, 若 p₁=n, 则 p_i=_____。
A、i B、n-i C、n-i+1 D、不确定
- 串的长度是_____。
A、串中不同字母的个数 B、串中不同字符的个数
C、串中所含字符的个数，且大于 0 D、串中所含字符的个数
- 数组 A 中，每个元素占用的存储空间长度为 3 个字节，行下标 i 从 0 到 7，列下标 j 从 0 到 9，从首地址 SA 开始连续存放在存储器内，该数组按行存放时，元素 A[7][4]的起始地址为_____。
A、SA+141 B、SA+144 C、SA+222 D、SA+225
- 在一非空二叉树的中序遍历序列中，根结点的右边()。
A、只有右子树上的所有结点 B、只有右子树上的部分结点
C、只有左子树上的部分结点 D、只有左子树上的所有结点
- Huffman 树的形态是_____。
A、唯一的 B、不定的 C、不唯一 D、以上说法都不对

8. 深度为 5 的二叉树至多有_____结点。
 A、16 B、32 C、31 D、10
9. 已知某二叉树的后序遍历序列是 dabec，中序遍历序列是 debac，它的前序遍历序列是_____。
 A、acbed B、decab C、deabc D、cedba
10. 在一个图中，所有顶点的度数之和等于所有边数的_____倍。
 A、1/2 B、1 C、2 D、4
11. 任何一个无向连通图的最小生成树_____。
 A、只有一棵 B、有一棵或多棵
 C、一定有多棵 D、可能不存在
12. 在图的表示法中，表示形式唯一的是_____。
 A、邻接矩阵表示法 B、邻接表表示法
 C、逆邻接表表示法 D、邻接表和逆邻接表表示法
13. 与其他查找方法相比，散列表查找法的特点是_____。
 A、通过关键字比较进行查找
 B、通过关键字计算记录存储地址进行比较
 C、通过关键字计算记录存储地址，并进行一定的比较进行查找
 D、按存储顺序查找
14. 用二分法在有序表{3, 4, 10, 13, 33, 42, 46, 63, 76, 78, 95, 96, 120}中查找 95 时，要进行的比较次数为_____。
 A、2 B、3 C、4 D、5
15. 堆是_____。
 A、完全二叉树 B、线性表 C、二叉排序树 D、平衡二叉树

二. 填空题（每空 3 分，本题共 30 分）

1. 算法的五个重要特性分别是有穷性、确定性、_____、输入和输出。
2. 指针 p 指向非空循环单链表 head 的尾结点，则 p 满足_____。
3. 有如下程序，则此程序的输出结果（其中队列的元素类型为 char）是_____。

```
void main()
{ char x='e', y='c';
  enqueue(q, 'h'); enqueue(q, 'r'); enqueue(q, y);
```

```

dequeue(q,x); enqueue(q,x);
dequeue(q,x); enqueue(q,'a');
while (!queueempty(q))
{ dequeue(q,y); printf(y);}
printf(x);
}

```

4. 设串 $s_1='ABCDEFGH'$, $s_2='PQRST'$, 函数 $Concat(x,y)$ 返回 x 串和 y 串的连接串, $Substr(s,i,j)$ 返回串 s 从序号 i 开始的 j 个字符组成的子串, $Length(s)$ 返回串 s 的长度, 则 $Concat(Substr(s_1,2,Length(s_2)),Substr(s_1,Length(s_2),2))$ 的结果串为_____。
5. 设广义表 $L=((),())$, 则 $GetTail(L)=$ _____。
6. 假定一棵树的广义表表示为 $A(B(C,D(E,F,G),H(I,J)))$, 则度为 3 的结点数有_____个。
7. 具有 n 个结点的二叉树中, 有_____个空指针。
8. 已知一棵深度为 6 的完全二叉树的第 6 层有 7 个叶子结点, 则该完全二叉树共有_____个结点。
9. 在无向图的邻接表存储结构中, 第 i 个链表中的结点个数是顶点 v_i 的_____。
10. _____法构造的哈希函数一定不会发生冲突。

三. 算法填空题 (每小题 4 空, 每空 3 分, 本题共 36 分。带下划线编号处为填写位置)

1. 设一个长度大于 1 的循环单链表中, 既无头结点也无头指针, p 为指向该链表中某个结点的指针, 以下为一个删除该结点直接前驱结点的算法。

```

void delete(LinkList p)
{
//在一个既无头结点也无头指针的长度大于一的循环链表中,
//删除指针 p 所指的某个结点的直接前驱结点
q=_____ (1) _____; //查找p结点的前驱结点q
while(q->next!=p)
    q=_____ (2) _____;
r=q; //查找 q 结点的前驱结点 r
while(_____ (3) _____)
    r=r->next;
}

```

```

r->next=_____ (4) _____;
free(q);
}

```

2. 设二叉树以二叉链表为存储结构，以下是一个对二叉树进行先序遍历的非递归算法。

```

Status preorder(BiTree t,Status(*Visit)(TelemType e))
{
//对以二叉链表为存储结构的二叉树，进行先序遍历的非递归算法
InitStack(s);
_____ (1) _____; //根结点入栈
while(_____ (2) _____) //栈非空
{
GetTop(s,p)
Visit(_____ (3) _____);
Pop(s,p);
if (p->rchild)
Push(s,p->rchild);
if(p->lchild)
Push(s,_____ (4) _____);
} //while
return OK;
} // preorder

```

3. 设图G是用邻接矩阵存储的，以下是一个从顶点 V_1 出发广度优先遍历图的算法。

```

void BFSTraverse(MGraph G,Status(*Visit)(int v))
{
// 对以数组存储表示的图 G 进行广度优先搜索遍历
bool visited[G.vexnum]; // 附设访问标识数组
Queue Q; // 附设队列 Q
for (v=0; v<G.vexnum; ++v) visited[v] = FALSE;
InitQueue(Q); // 设置空队列 Q
for ( v=0; v<_____ (1) _____; ++v )
if (_____ (2) _____)
{ // 从每一个未被访问的顶点出发进行广度优先搜索

```

```

visited[v] = TRUE;    Visit ( _____ (3) _____ ); // 访问图中第 v 个顶点
EnQueue(Q, v);      // v 入队列
while (!QueueEmpty(Q)) {
    DeQueue(Q, u);   // 队头元素出队并置为 u
    for ( w=0; w<G.vexnum; w++; )
        if ( G.arcs[u, w].adj && _____ (4) _____ )
            {
                visited[w] = TRUE;    Visit(w);    // 访问图中第 w 个顶点
                EnQueue(Q, w);      // 当前访问的顶点 w 入队列 Q
            } // if
    } // while
} // if
DestroyQueue(Q);
} // BFSTraverse

```

四. 算法设计题 (每小题 12 分, 本题共 36 分)

1. 有两个顺序表 A (有 m 个元素) 和 B (有 n 个元素), 其元素均按关键字从小到大顺序排列, 写算法将它们合并为一个表 C, 使表 C 仍按关键字从小到大顺序排列。
2. 已知两个头指针为 pa 和 pb 的单链表分别表示两个集合, 其元素按递增有序, 写算法求出两个集合的交集 pc, pc 同样以递增有序的单链表形式存放。
3. 设一棵二叉树以二叉链表为存储结构, 写一个计算二叉树中所有结点数的算法。

五. 混合题 (其中第 1 小题 8 分, 第 2 小题 10 分, 本题共 18 分)

1. 选取哈希函数 $H(k) = (3k) \text{ MOD } 11$ 。用线性探测开放定址法处理冲突, 试在 0~10 的散列地址空间中对关键字序列 (22, 41, 53, 46, 30, 13, 01, 67) 构造哈希表。
2. 以关键字序列 (48、36、68、99、75、24、28、52) 为例, 手工执行以下排序算法 (升序):

(1) 希尔排序 (增量序列为 5, 3, 1)

(2) 快速排序