

华北水利水电学院 2005 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) 可以先用文字简述算法解题思路; (2) 给出数据存储结构定义; (3) 用类 C 或类 Pascal 语言写出算法; (4) 算法中应给出适当注释。

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、线性结构和非线性结构 D、内部结构和外部结构
- 以下关于线性表的说法不正确的是()。
A、线性表中的数据元素可以是数字、字符、记录等不同类型。
B、线性表中包含的数据元素个数不是任意的。
C、线性表中的每个结点都有且只有一个直接前趋和直接后继。
D、存在这样的线性表: 表中各结点都没有直接前趋和直接后继。
- 若已知一个栈的入栈序列是 1, 2, 3, ..., n, 其输出序列是 $p_1, p_2, p_3, \dots, p_n$, 若 $p_1=n$, 则 $p_i=()$ 。
A、i B、n-i C、n-i+1 D、不确定
- 在计算递归函数时, 如不使用递归过程, 则一般情况下必须借助于()数据结构。
A、栈 B、树 C、双向队列 D、广义表
- 深度为 5 的二叉树至多有 () 结点。
A、16 B、32 C、31 D、10
- Huffman 树的形态是()。
A、唯一的 B、不定的 C、不唯一的 D、以上说法都不对
- 非空二叉树的前序序列和后序序列正好相反, 则二叉树一定是 () 的二叉树。
A、空或只有一个结点 B、高度等于其结点数
C、任一结点无左孩子 D、任一结点无右孩子
- 已知某二叉树的后序遍历序列是 dabec, 中序遍历序列是 debac, 它的先序遍历序列是 ()。
A、acbed B、decab C、deabc D、cedba
- 树的基本遍历策略分为先根遍历和后根遍历; 二叉树的基本遍历策略可分为先序遍历、中序遍历和后序遍历。结论()是正确的。
A、树的先根遍历序列与其对应的二叉树的先序遍历序列相同
B、树的后根遍历序列与其对应的二叉树的先序遍历序列相同
C、树的先根遍历序列与其对应的二叉树的中序遍历序列相同
D、以上都不对

10. 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
11. ()适合用邻接表表示。
- A、稠密图 B、有向完全图
C、无向完全图 D、稀疏图
12. 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
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,并通过一定的比较进行查找
D、按存储顺序查找
14. 冒泡排序对关键字序列{18, 16, 14, 12, 10, 8}进行从小到大的排序,所要进行的关键字比较总次数为()。
- A、10 B、15 C、21 D、34
15. 对某无序元素序列进行快速排序时,每次划分选择的基准元素为该序列的 ()时,得到的两个子区间是均匀的。
- A、最大值 B、最小值
C、任意值 D、中间值

二. 填空题 (每小题 3 分, 本题共 30 分)

1. 已知 L 是无头结点的单链表,试从下列提供的答案中选择合适的语句序列,实现在表首插入 S 结点的语句序列是_____。
- ①P->next=S; ②P=L;
③L=S; ④P->next=S->next;
⑤S->next= P->next; ⑥S->next=L;
⑦S->next=NULL; ⑧while (P->next!=Q) P=P->next;
⑨while (P->next!=NILL) P=P->next;
2. 设栈 S 和队列 Q 的初始状态为空,元素 a、b、c、d、e、f 依次通过栈 S,一个元素出栈后即进入队列 Q。若这 6 个元素出队列的顺序是 b、d、c、f、e、a,则栈 S 的容量至少应该是_____。
3. 设串 s1='ABCDEFGF', s2='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(s1,2,Length(s2)),Substr(s1,Length(s2),2))的结果串为_____。
4. 二维数组 A[10][5]采用行序为主序方式存储,每个元素占 4 个存储单元,并且 A[5][3]的存储地址是 1000,则 A[8][2]的地址是_____。
5. 设广义表 A=(x,((a,b),c,d)),则 Head(Head(Tail(A)))=_____。
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，则该完全二叉树总共有_____个叶子结点。

9. 设有一个有序文件，各记录的关键字为{2,3,5,7,11,13,17,19,23,29,31,37,41,43,47}，当用折半查找算法查找关键字为 7 的记录时，比较次数为_____。

10. 用_____法构造的哈希函数一定不会发生冲突。

三. 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，写一个删除该结点直接前驱结点的算法。(本题 12 分)

四. 设有头指针为 $head$ 的单链表，写算法在链表中查找出所有按先后顺序出现的元素 x 和 y ，并将 x 和 y 之间的所有结点（不含 x 和 y ）全部删除之。(本题 12 分)

五. 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全部叶子结点数的算法。(本题 12 分)

六. 算法填空（填空位置在下划线处）：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中序遍历的非递归算法。（每空 3 分，本题共 12 分）

```
Status inordertraverse(BiTree t, Status(*Visit)(TelemType e)) {
    //对以二叉链表为存储结构的二叉树 t，采用非递归算法进行中序遍历
    InitStack(s);
    _____ (一) _____; //根指针进栈
    while(!StackEmpty(s))
    { while (_____ (二) _____ &&p)
        Push(s,p->lchild); //向左走到尽头
        _____ (三) _____; //空指针退栈
        if (!StackEmpty(s))
        { Pop(s,p);
            if(!_____ (四) _____)
                return error;
            Push(s,p->rchild);
        } //if
    } //while
    return OK;
} //inordertraverse
```

七. 算法填空（填空位置在下划线处）：以下是一个建立有向图邻接表的算法，结点偶对由用户输入，以 (0, 0) 结束，每条边上无信息。（每空 3 分，本题共 12 分）

```
Status CreateDG(ALGraph &G) {
    //采用邻接表表示法，构造有向图 G
    scanf (&G. vexnum, &G. arcnum);
```

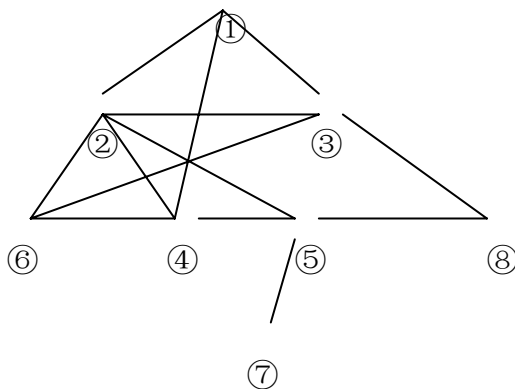
```

for (i=0;i<_____ (一) _____; ++i) //建立顶点信息
    {scanf(&G.vertices[i].data); G.vertices[i].firstarc=NULL;}
scanf(&v1,&v2); //输入一个偶对
while(v1!=0&&v2!=0) //建立邻接表
    { i=LocateVex(G,v1); j= LocateVex(G,v2);
      s=(ArcNode *)malloc(sizeof(ArcNode));
      s->adjvex=_____ (二) _____;
      s->nextarc= G.vertices[i].firstarc;
      G.vertices[i].firstarc=_____ (三) _____;
      scanf(_____ (四) _____);
    }
} //CreateDG

```

八. 证明题: 任意一棵有 n 个结点的二叉树, 已知它有 m 个叶子结点, 试证明非叶子结点中有 $(m-1)$ 个度为 2, 其余度数为 1。(本题 10 分)

九. 用深度优先搜索和宽度优先搜索对下图所示的无向图进行遍历(从顶点 1 出发), 给出遍历序列。(本题 10 分)



十. 以关键码序列 (50,08,51,06,90,17,89,27,50,42) 为例, 按升序手工执行以下排序算法: (每小题 5 分, 本题共 10 分)

- (1) 希尔排序 (增量序列为 5, 3, 1) (2) 快速排序。