

科目代码: 896

科目名称: 数据结构

北京工业大学 2010 年专业学位硕士研究生入学考试试题

★所有答案必须做在答题纸上, 做在试题纸上无效!

一、单项选择题 (共 20 分, 每小题 2 分)

下列每个题目中有四个选项, 其中只有一个是正确的。请根据题目的陈述, 选择正确的答案。

1. 数据结构是指 ()。

- A. 数据的基本单位
- B. 性质相同的数据元素的集合
- C. 相互之间存在一种或多种特定关系的数据元素的集合
- D. 描述客观事物且由计算机处理的数值、字符等符号的总称

2. 算法的时间复杂度是指 ()。

- A. 算法执行的绝对时间
- B. 随着问题规模 n 的增大, 算法执行时间的增长趋势
- C. 算法中执行语句的条数
- D. 获得算法执行时间的复杂程度

3. 如果入栈序列为 ABCD, 不可能出现的出栈序列是 ()。

- A. ABCD
- B. DABC
- C. ACDB
- D. DCBA

4. 队列是一种特殊的线性表, 其特殊性在于 ()。

- A. 只能插入
- B. 插入、删除在同一端
- C. 只能删除
- D. 插入在一端, 删除在另一端

5. 如果对任意一棵二叉树进行后序遍历, 根结点位于后序遍历序列的 ()。

- A. 第一个
- B. 最中间
- C. 最后一个
- D. 不确定

科目代码:

896

科目名称:

数据结构

★所有答案必须做在答题纸上, 做在试题纸上无效!

6. 对线性表进行二分查找时, 要求线性表必须 ()。
- A. 采用顺序存储结构方式存储, 对元素内容排列无要求
 - B. 采用链式存储结构方式存储, 对元素内容排列无要求
 - C. 采用顺序存储结构方式存储, 且要求元素内容排列有序
 - D. 采用链式存储结构方式存储, 且要求元素内容排列有序
7. 无向图的最小生成树是指 ()。
- A. 利用深度优先遍历得到的生成树
 - B. 利用广度优先遍历得到的生成树
 - C. 权值之和最小的生成树
 - D. 边的条数最少的生成树
8. 在二叉排序树中, 数值最小的结点是 ()。
- A. 最左结点
 - B. 最右结点
 - C. 根结点
 - D. 不确定
9. 有一种排序方法, 它使用的基本排序手段是插入, 适用于原始数据的排列基本有序且数据量 n 较小的情况, 这种排序方法是 ()。
- A. 简单选择排序
 - B. 冒泡排序
 - C. 直接插入排序
 - D. 快速排序
10. 在下列 4 种排序算法中, 时间复杂度为 $O(n \log n)$ 的排序算法是 ()。
- A. 简单选择排序
 - B. 冒泡排序
 - C. 直接插入排序
 - D. 堆排序

科目代码: 896

科目名称: 数据结构

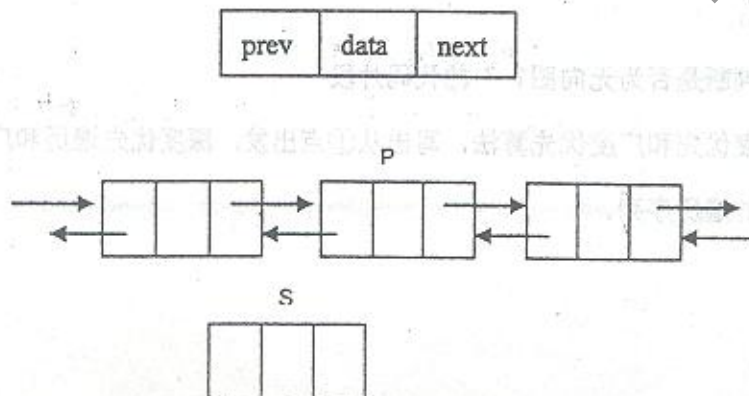
★所有答案必须做在答题纸上, 做在试题纸上无效!

二、填空题 (20 分, 每小题 2 分)

1. 在采用顺序存储结构表示队列时, 通常将其设计为循环队列, 主要原因是 (1) 。
2. 广义表 $GL = (((((a), b), d), e))$ 的深度是 (2) 。
3. 已知两个递增有序的单链表 (长度分别为 m 和 n), 如果将其归并成一个按元素递增有序的单链表, 可能达到的最优时间复杂度为 (3) 。
4. 深度为 h 的完全二叉树至少应该包含的结点个数为 (4) 。
5. 对于任何一棵二叉树 T , 如果终端 (叶子) 结点的数目为 n_0 , 度为 2 的结点数目为 n_2 , 则 $n_0 =$ (5) 。
6. 赫夫曼树是指 (6) 。
7. 所谓树的遍历是指 (7) 。
8. 衡量查找算法时间性能的指标是 (8) 。
9. 时间复杂度为 $O(n \log n)$, 空间复杂度为 $O(1)$ 的排序算法是 (9) 。
10. 堆排序使用的基本排序手段是 (10) 。

三、解答题 (50 分, 每小题 10 分)

1. 试写出将 S 结点插入到双向链表中 P 结点之前的语句序列。结点结构为:



科目代码: 896

科目名称: 数据结构

★所有答案必须做在答题纸上, 做在试题纸上无效!

2. 已知一棵二叉树的先序遍历序列和中序遍历序列分别为:

先序遍历序列 ABCDEFGHIJ

中序遍历序列 CBEDAGHFJI

完成下列问题:

- (1) 画出这棵二叉树
- (2) 写出这棵二叉树的后序遍历序列
- (3) 写出这棵二叉树的深度

3. 已知图G含有8个顶点, 它们之间的邻接关系用下列邻接矩阵Adj表示。

	①	②	③	④	⑤	⑥	⑦	⑧
①	0	0	0	1	1	1	0	0
②	0	0	1	0	0	0	1	0
③	0	1	0	0	0	0	0	1
④	1	0	0	0	1	0	0	0
⑤	1	0	0	1	0	1	0	0
⑥	1	0	0	0	1	0	0	0
⑦	0	1	0	0	0	0	0	0
⑧	0	0	1	0	0	1	0	0

完成下列问题:

- (1) 写出“判断是否为无向图?”的代码片段
- (2) 按照深度优先和广度优先算法, 写出从①点出发, 深度优先遍历和广度优先遍历这个图的遍历序列。

科目代码: 896

科目名称: 数据结构

★所有答案必须做在答题纸上, 做在试题纸上无效!

4. 对于整数序列 (498, 012, 507, 903, 170, 892, 270, 648, 607, 149), 构造一棵二叉排序树, 画出构造结果, 并计算平均查找长度 ASL。

5. 根据字母集 {s, t, a, e, d} 中每个字母在电文中出现的频度所构建的赫夫曼编码为:

s:	101
t:	00
a:	01
e:	11
d:	100

完成下列问题:

(1) 画出对应的赫夫曼树

(2) 假设接收到的电文为 110100100010001, 根据赫夫曼编码给出译文

计算机/软件工程专业

每个学校的

考研真题/复试资料/考研经验

考研资讯/报录比/分数线

免费分享



微信 扫一扫

关注微信公众号

计算机与软件考研

科目代码: 896

科目名称: 数据结构

★所有答案必须做在答题纸上, 做在试题纸上无效!

四、算法阅读题 (20 分, 每小题 5 分)

根据所给二叉树的实例, 阅读算法 algorithm, 并回答问题。

```
void algorithm (BinTree T) {
```

```
    // T 为指向二叉树根结点的指针
```

```
    Stack S;
```

```
    BinTree p, q;
```

```
    If (T == NULL) return;
```

```
    InitStack( S );
```

```
    p=T;
```

```
    do {
```

```
        while (p) {
```

```
            push( S, p );
```

```
            if ( p->lchild ) p = p->lchild;
```

```
            else p = p->rchild;
```

```
        }
```

```
        while( ! StackEmpty(S) && q = getTop(S) && q->rchild == p ) {
```

```
            p = pop(S);
```

```
            printf("%c", p->data);
```

```
        }
```

```
        if (!StackEmpty(S)) { // ----- ②
```

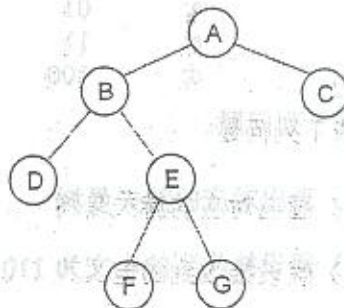
```
            q = getTop(S);
```

```
            p = q->rchild;
```

```
        }
```

```
    } while (!StackEmpty(S));
```

```
}
```



1. 写出标号①所对应 while 语句的功能。
2. 写出标号②所对应 if 语句的功能。
3. 写出算法的运行结果。
4. 说明整个算法的功能。

科目代码: 896

科目名称: 数据结构

★所有答案必须做在答题纸上, 做在试题纸上无效!

五、算法设计题 (40 分, 第 1 小题 25 分, 第 2 小题 15 分)

按照题目要求设计算法。

1. 已知一棵二叉树 T 。试设计一个计算二叉树高度的递归算法。

假设二叉树采用链式存储结构存储, 其结点结构的类型定义为:

```
typedef struct BiTNode{  
    int data;  
    struct BiTNode *lchild, *rchild;  
} Node, *BTree;
```

实现该算法的函数原型定义为:

```
int BinTreeHigh(BTree tree)
```

其中, $tree$ 是指向二叉树根结点的指针。

2. 设有一个仅由红、白、蓝 3 种颜色的色块组成的序列, 各种色块的个数是随机的, 原始排列的顺序也是任意的, 但是 3 种颜色色块的总数为 n 。试设计一个时间复杂度为 $O(n)$, 空间复杂度为 $O(1)$ 的算法, 使得这些色块按照红、白、蓝的顺序排列, 即排成类似荷兰国旗的红白蓝三块色条的图案。要求: 首先设计数据结构, 然后选择存储结构, 最后再设计算法。