

北京工业大学 2019——2020 学年第 2 学期

《计算机软件基础》 期末考试试卷 A 卷

考试说明：考试时间：95 分钟

考试形式：在线开卷

适用专业：电子信息工程、通信工程、自动化、机器人工程、电子科学与技术、微电子工程

本人已学习了《北京工业大学考场规则》和《北京工业大学学生违纪处分条例》，承诺在考试过程中自觉遵守有关规定，服从监考教师管理，确保整个考试过程均在摄像头可视范围之内且监控不中断，不对试题进行截屏、拍照等，不通过手机、QQ 等各种手段向他人寻求答案；若有违反，愿接受相应的处分。

阅读完毕后请将以下文字誊抄在答题纸首页，并做好答题准备。

本人已认真阅读以上要求，知晓相关规定并遵守执行，若有违反，愿接受相应的处分。

承诺人：\_\_\_\_\_ 学号：\_\_\_\_\_ 班号：\_\_\_\_\_

注：本试卷共 4 大题，共 8 页，满分 100 分。并将答案写在答题纸上，如因答案写在其他位置而造成的成绩缺失由考生自己负责。

卷 面 成 绩 汇 总 表（阅卷教师填写）

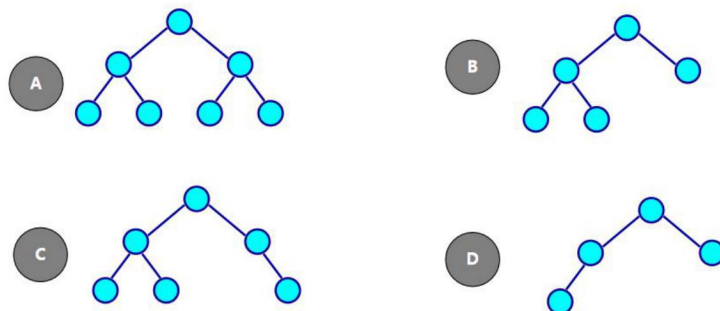
题号	一	二	三	四	总成绩
满分	15	11	20	54	
得分					

一、 单选题（15 分，每小题 1.5 分，共 10 小题）

1. 从逻辑上可以把数据结构分成（ C ）。

- A. 动态结构和静态结构                      B. 顺序结构和链式结构  
C. 线性结构和非线性结构                      D. 初等结构和组合结构

2. 下列二叉树中，（ C ）不是完全二叉树。



3. 为了便于衡量算法效率，突出算法的特点，屏蔽编译器等因素的差异，使算法的时间性能仅依赖于（ A ）。

- A. 硬件速度      B. 编程语言      C. 代码质量      D. 问题规模

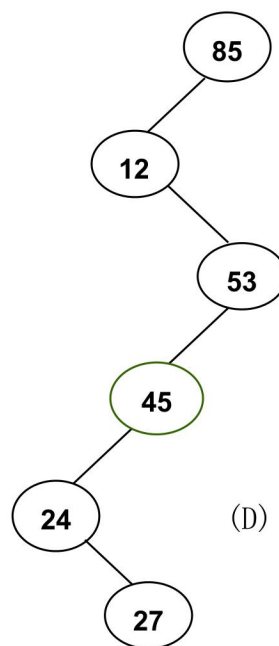
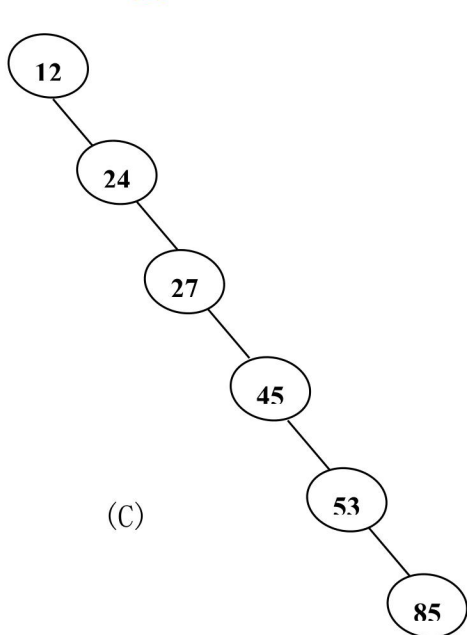
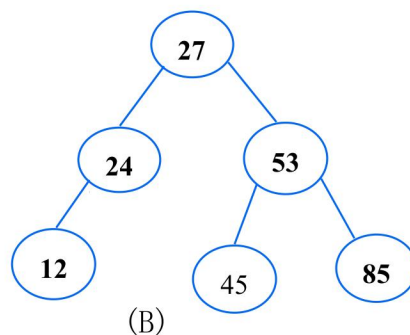
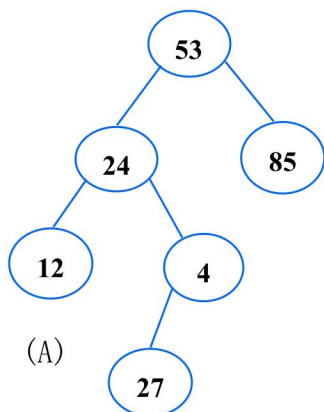
4. 给定原始序列：(29, 38, 12, 24, 7, 19, 59, 35, 47)，使用直接插入排序方法从小到大进行排序，进行到第 4 趟时的排序结果是（ A ）。

- A. 7, 12, 24, 29, 38, 19, 59, 35, 47  
B. 7, 12, 19, 29, 38, 24, 59, 35, 47  
C. 7, 12, 19, 24, 29, 38, 59, 35, 47  
D. 7, 12, 19, 24, 38, 59, 29, 35, 47

5. 信箱通信是一种 ( ) 方式。

- A. 直接通信      B. 高级通信      C. 低级通信      D. 管道通信

6. 对一组关键字序列 (53, 24, 45, 85, 27, 12), 以不同的先后次序生成如下四种不同形态的二叉排序树, 其中 ASL 最小的是 ( B )。



7. 下列有关软件测试和软件调试的论述, 错误的是 ( C )。

A. 软件测试是找出软件已经存在的错误, 而调试是定位错误, 修改程序以修正错误。

B. 测试经常是由独立的测试人员在不了解软件设计的条件下完成的; 调试必须由了解详细设计的开发人员完成。

C. 调试是在测试之前, 在方法、思路、策略上都有所不同。

D. 软件测试可以计划，可以预先制定测试用例和过程，工作进度可以度量。而调试不能计划，进度不可度量。

8. 设串  $s1 = 'BCDEFG'$ ， $s2 = 'PQRST'$ ，函数  $con(x, y)$  返回  $x$  和  $y$  串的连接串， $subs(s, i, j)$  返回串  $s$  的从序号  $i$  的字符开始的  $j$  个字符组成的子串， $len(s)$  返回串  $s$  的长度，则  $con(sub(s1, 2, len(s2)), sub(s1, len(s2), 2))$  的结果是 ( )。

- A. 'BCDEF'    B. 'BCDEFG'    C. 'BCPQRST'    D. 'CDEFGFG'

9. 邻接表存储方法是一种 (A) 相结合的存储方法。

- A. 顺序存储和链式存储    B. 链式存储和索引存储  
C. 索引存储和散列存储    D. 散列存储和顺序存储

10. 栈  $S$  最多能容纳 4 个元素。现有 6 个元素按 A、B、C、D、E、F 的顺序进栈，问下列哪一个序列 (C) 是可能的出栈序列？

- A. E D C A B F    B. B C E F A D    C. C B E D A F    D. A D F E B C

## 二、 判断题 (11 分. 每小题 1 分共 11 小题)

- (X) 度为 2 的树是二叉树。
- (V) 如果某种排序算法是不稳定的，则该排序方法没有实际应用价值。
- (X) 由一棵二叉树的先序序列和后序序列可以唯一确定它。
- (V) 数据结构的选择或设计属于软件生命周期中开发期的工作。
- (V) 进程和程序是一一对应的，一个程序对应一个进程。
- (V) 三元组表是对稀疏矩阵进行压缩存储的一种方式。
- (X) 衡量算法性能的时间复杂度的单位是秒。
- (V) 邻接矩阵适用于稠密图，而邻接表适用于稀疏图。
- (V) 串是一种特殊的线性表，其特殊性体现在数据元素只能是字符。
- (V) 栈结构常采用的两种存储结构是线性存储结构和非线性存储结构。
- (V) 在散列查找中， $ASL=0$ ，因而也没有‘比较’操作。

## 三、 填空题 (20 分. 每小题 2 分, 共 10 小题)

1. 已知一棵完全二叉树中共有 768 个结点, 则该树中共有  $\frac{768}{2} = 384$  个叶子结点。
2. 在存储串时, 若串较小, 则采用 数组 存储方式最为节省空间。
3. 在有序表 (2, 11, 16, 22, 25, 27, 33, 39, 42, 56, 77, 79, 81) 中采用折半查找方法查找给定值 13 需要依次经过与关键字 33, 16, 11, 2 比较才能确定有序表中无此给定值。
4. 具有  $n$  个结点的完全二叉树, 其编号最小的叶子结点序号为  $\lfloor \frac{n}{2} \rfloor + 1$ 。
5. 对于一个栈, 如果输入序列依次为 a, b, c, 则全部可能的输出序列为 abc, bac, bca, cba。
6. 假设有二维数组 A[6, 8], 每个元素用相邻的 6 个字节存储, 存储器按字节编址, 若 A[0][0] 的起始存储位置 (基地址) 为 1000, 则按列存储时, 元素 A[4][7] 的第一个字节的地址为  $1000 + (6 \times 7 + 4) \times 6 = 1784$ 。
7. 设循环队列的容量为 40 (序号从 0 到 39), 现经过一系列的入队和出队运算后, 有 front=19, rear=11, 则循环队列中有元素 32 个。
8. 在单处理器系统中, 如果同时存在 11 个进程, 则处于就绪队列中的进程最多有 10 个。
9. 对于  $n$  个顶点  $e$  条边的无向图 G, 假定每个顶点占 1 个存储空间, 且不考虑指针所占空间, 则采用邻接矩阵表示所需的存储空间为  $n^2$ , 采用邻接表表示所需的存储空间为  $2e$ 。
10. 若对一组纪录 (46, 79, 56, 38, 40, 80, 35, 50, 74) 进行直接插入排序, 当把第 9 个纪录插入到前面已排序的有序表时, 为寻找插入位置需比较 3 次。

## 四、 综合题 (54 分. 第小题 6 分)

1. 阅读下列程序, 说出该算法的功能, 并给出算法的时间复杂度。

```
typedef int ElemType;
typedef struct {
    ElemType data[MaxSize];
    int length;
} SqList;
```

资料由公众号【工大喵】收集整理并免费分享

```
void test1( SqList *&L ){
ElemType temp;
int i,j=L->length-1;
while(i<j){
    while (L->data[i]<0)
        i++;
    while (L->data[j]>0)
        j--;
    if(i<j){
        temp = L->data[i];
        L->data[i]=L->data[j];
        L->data[j]=temp;
    }
}
}
```

2. 阅读下列程序，说出该算法的功能，已知 L 为带头结点的单链表。

```
typedef int ElemType;
typedef struct LNode {
ElemType data;
struct LNode *next;
} NODE;
void test2 ( NODE *&L, ElemType x){
NODE *p = L->next, *pre = L;
NODE *xp = p, *xpre = L,*s;
while ( p!=NULL){
if (xp->data < p->data){
    xp = p;
    xpre = pre;
}
}
```

资料由公众号【工大喵】收集整理并免费分享

```

pre = p; p = p->next;
}
s = (NODE *)malloc( sizeof(NODE) );
s->data = x;
s->next = xpre->next;
xpre->next = s;
}

```

3. 线性表有两种存储结构：一是顺序表，二是链表。试问：

(1) 如果有  $n$  个线性表同时并存，并且在处理过程中各表的长度会动态变化，线性表的总数也会自动地改变。在此情况下，应选用哪种存储结构？为什么？

(2) 若线性表的总数基本稳定，且很少进行插入和删除，但要求以最快的速度存取线性表中的元素，那么应采用哪种存储结构？为什么？

4. 已知一棵二叉树的中序、后序序列分别为：

中序 D A E F B H G N K J L I M

后序 D F E A H G B K L J M I N

求：(1) 画出该二叉树

(2) 写出先序序列

5. 某通讯系统只使用 8 种字符 a、b、c、d、e、f、g、h，其使用频率分别为 0.12, 0.13, 0.21, 0.07, 0.18, 0.02, 0.10, 0.17。试求：

1. 构造哈夫曼树

2. 设计哈夫曼编码

3. 求出其 wpl 值

6. 对如下关键字序列构建哈希表，地址空间为 0-16。

(zhao, qian, sun, li, zhou, wu, zheng, wang, feng, chen, chu, wei)

哈希函数为  $H(K) = (\text{Ord}(\text{关键字 } k \text{ 的第一个字母}) - \text{Ord}('a') + 1) / 2$ ， $\text{Ord}(x)$  表示  $x$  的 ASCII 码值，如： $\text{Ord}('a')$  为字母 'a' 的 ASCII 码值 97。



回答下列问题：

- (1) 用线性探测开放定址法处理冲突，画出构造的哈希表。
- (2) 写出查找关键字“zheng”时，在查找过程中依次进行比较的关键字序列。
- (3) 假定查找每个关键字的概率相等，计算该哈希表查找成功时的平均查找长度 ASL。

7. 已知一个图的顶点集  $V$ 、各边集  $E$  如下：

$$V = \{0, 1, 2, 3, 4, 5, 6\};$$

$$E = \{(0, 1), (0, 4), (1, 2), (1, 5), (2, 4), (3, 4), (3, 6), (4, 5), (5, 6)\}$$

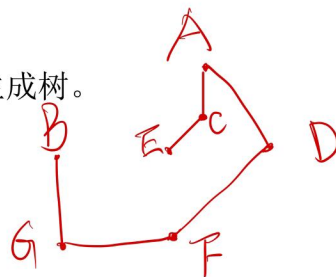
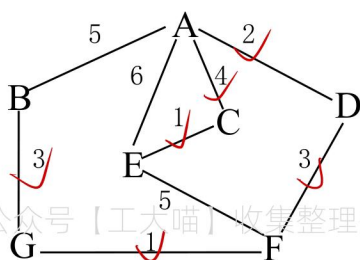
(1) 当它用邻接表表示时，分别写出从顶点  $V_0$  出发按深度优先搜索遍历得到的顶点序列和按广度优先搜索遍历得到的顶点序列。假定每个顶点邻接表中的结点是按顶点序号从大到小的次序链接的。

(2) 当它用邻接矩阵表示时，分别写出从顶点  $V_0$  出发按深度优先搜索遍历得到的顶点序列和按广度优先搜索遍历得到的顶点序列。

8. 有一种简单的排序算法，叫做计数排序 (count sorting)。这种排序算法对一个待排序的表 (用数组表示) 进行排序，并将排序结果存放到另一个新的表中。设表中所有待排序的关键字的值互不相同，计数排序算法针对表中的每个记录，扫描待排序的表一趟，统计表中有多少个记录的关键字比该记录的关键字小，假设针对某一个记录，统计出的计数值为  $d$ ，那么，这个记录在新的有序表中的合适的存放位置即为  $d$ 。问：

- 1) 对于有  $n$  个记录的表，关键字的比较次数是多少?
- 2) 与简单选择排序相比较，这种方法是否更好?为什么?

9. 考虑下图, 根据普利姆 (Prim) 算法, 求它的最小生成树。



资料由公众号【工大喵】收集整理并免费分享