

2018 年华中科技大学 834 计算机专业基础综合

复习八套卷 四 (🐸版)

参考答案

一. 填空题 (20 分, 每题 2 分)

1. 顺序存储结构是通过物理上相邻表示元素之间的关系的;链式存储结构是通过指针表示元素之间的关系的。
2. 已知指针 p 指向单链表 L 中的某结点, 则删除其后继结点的语句是: $u=p->next;$
 $p->next=u->next; free(u);$ 。
3. 设数组 A[0..8, 1..10], 数组中任一元素 A[i, j] 均占内存 48 个二进制位, 从首地址 2000 开始连续存放在主内存里, 主内存字长为 16 位, 那么设数组 A[0..8, 1..10], 数组中任一元素 A[i, j] 均占内存 48 个二进制位, 从首地址 2000 开始连续存放在主内存里, 主内存字长为 16 位, 那么数组按列存储时, 元素 A[5, 8] 的起始地址是 2204。
4. 一棵树 T 中, 包括一个度为 1 的结点, 两个度为 2 的结点, 三个度为 3 的结点, 四个度为 4 的结点和若干叶子结点, 则 T 的叶结点数为 21。
5. 如果结点 A 有 3 个兄弟, 而且 B 是 A 的双亲, 则 B 的度是 4。
6. G 是一个非连通无向图, 共有 28 条边, 则该图至少有 9 个顶点。
7. 换式网络是以交换器为中心构造的网络体系, 它们工作在 OSI 参考模型的第二层, 它们在不同的端口之间传递数据。
8. 文件传输协议的英文缩写为 FTP。
9. 由于帧中继可以不用网络层而使用链路层来实现复用和转接, 所以帧中继通信节点的层次结构中只有物理层和链路层。
10. 常用的 IP 地址有 A、B、C 三类, 128.11.3.31 主机标识 (hosted) 为 3.31。

二. 判断题 (20 分, 每个 2 分)

No.	1	2	3	4	5	6	7	8	9	10
Answer	×	×	×	×	×	√	√	×	√	√

1. 若一棵二叉树中的结点均无右孩子, 则该二叉树的中根遍历和后根遍历序列正好相反。
(×)
2. 对具有 n 个结点的堆进行插入一个元素运算的时间复杂度为 $O(n)$ 。(×)
3. 在散列法中采取开散列 (链地址) 法来解决冲突时, 其装载因子的取值一定在 (0, 1) 之间。(×)
4. 图的深度优先搜索是一种典型的回溯搜索的例子, 可以通过递归算法求解。(×)

5. 对任何用顶点表示活动的网络（AOV 网）进行拓扑排序的结果都是唯一的。（×）
6. 邻接矩阵适用于稠密图（边数接近于顶点数的平方），邻接表适用于稀疏图（边数远小于顶点数的平方）。（√）
7. ISO 划分网络层次的基本原则是：不同节点具有相同的层次，不同节点的相同层次有相同的功能。（√）
8. 所有以太网交换机端口既支持 10BASE-T 标准，又支持 100BASE-T 标准。（×）
9. RIP（Routing Information Protocol）是一种路由协议。（√）
10. 在局域网标准中共定义了四个层。（√）

三. 选择题（30 分，每个 3 分）

No.	1	2	3	4	5	6	7	8	9	10
Answer	C	C	A	D	C	B	C	B	C	B

1. 某算法的时间复杂度为 $O(n^2)$ ，表明该算法的（ C ）。
 A. 问题规模是 n^2
 B. 执行时间等于 n^2
 C. 执行时间与 n^2 成正比
 D. 问题规模与 n^2 成正比
2. 静态链表中指针表示的是（ C ）。
 A. 内存地址 B. 数组下标
 C. 下一元素地址 D. 左、右孩子地址
3. 已知有向图 $G=(V, E)$ ，其中 $V=\{V_1, V_2, V_3, V_4, V_5, V_6, V_7\}$ ，
 $E=\{\langle V_1, V_2 \rangle, \langle V_1, V_3 \rangle, \langle V_1, V_4 \rangle, \langle V_2, V_5 \rangle, \langle V_3, V_5 \rangle, \langle V_3, V_6 \rangle, \langle V_4, V_6 \rangle, \langle V_5, V_7 \rangle, \langle V_6, V_7 \rangle\}$ ，G 的拓扑序列是（ A ）。
 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. $V_1, V_2, V_5, V_3, V_4, V_6, V_7$
4. 下列排序算法中，其中（ D ）是稳定的。
 A. 堆排序，冒泡排序 B. 快速排序，堆排序
 C. 直接选择排序，归并排序 D. 归并排序，冒泡排序
5. 若以 1234 作为双端队列的输入序列，则既不能由输入受限的双端队列得到，也不能由输出受限的双端队列得到的输出序列是（ C ）。
 A. 1234 B. 4132 C. 4231 D. 4213
6. 下列关于 AOE 网的叙述中，不正确的是（ B ）。
 A. 关键活动不按期完成就会影响整个工程的完成时间
 B. 任何一个关键活动提前完成，那么整个工程将会提前完成
 C. 所有的关键活动提前完成，那么整个工程将会提前完成
 D. 某些关键活动提前完成，那么整个工程将会提前完成
7. TCP / IP 网络中，某主机的 IP 地址为 130.25.3.135，子网掩码为 255.255.255.192，那么该主机所在的子网的网络地址是（ C ）。

- A. 130.25.0.0
- B. 130.25.3.0
- C. 130.25.3.128
- D. 130.25.3.255

【解析】题中子网掩码的前三个字节用二进制表示时全为“1”，第四个字节转换为二进制后为 11000000，使用子网掩码的二进制形式和 IP 地址的二进制表示形式相与后得到该 IP 地址所在子网的网络地址为

130.25.3.128。

8. IP 数据报的报文格式如下图所示。在没有选项和填充的情况下，报头长度域的值为（ B ）。

版本	报头长度	服务类型	总长度			
标识			标志	片偏移		
生存周期		协议	头部校验和			
源 IP 地址						
目的地 IP 地址						
选项 + 填充						

- A. 3
- B. 5
- C. 10
- D. 20

【解析】本题考查 IP 数据报的格式，计算报文头长度时，以 4 字节为单位，不含选项和填充的报文正好是

IP 数据报的固定部分，共 20 个字节，因此，报头长度为 $20 / 4 = 5$ 。

9. 分组交换对报文交换的主要改进是（ C ）。

- A. 差错控制更加完善
- B. 路由算法更加简单
- C. 传输单位更小且有固定的最大长度
- D. 传输单位更大且有固定的最大长度

【解析】相对于报文交换而言，分组交换中将报文划分为一个个具有固定最大长度的分组，以分组为单位进行传输。

10. 波特率等于（ B ）。

- A. 每秒钟传输的比特
- B. 每秒钟可能发生的信号变化的次数
- C. 每秒传输的周期数
- D. 每秒传输的字节数

【解析】波特率是指数据信号对载波的调制速率，它用单位时间内载波调制状态改变的次数来表示，其单位是波特（Baud）。波特率与比特率的关系是比特率 = 波特率 * 单个调制状态对应的二进制位数。

四. 简答题（60 分）

1. 用一维数组存放的一棵完全二叉树如下图所示：

A	B	C	D	E	F	G	H	I	J	K	L
---	---	---	---	---	---	---	---	---	---	---	---

写出后序遍历该二叉树时访问结点的顺序

答：HIDJKEBLFGCA

2. 判断下列序列是否是堆（可以是小堆，也可以是大堆，若不是堆，请将它们调整为堆）。

(1) 100, 85, 98, 77, 80, 60, 82, 40, 20, 10, 66

(2) 100, 98, 85, 82, 80, 77, 66, 60, 40, 20, 10

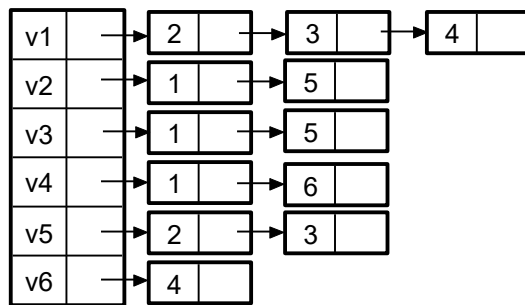
(3) 100, 85, 40, 77, 80, 60, 66, 98, 82, 10, 20

答：(1)是大堆；

(2)是大堆；

(3)不是堆，调成大堆 100, 98, 66, 85, 80, 60, 40, 77, 82, 10, 20

3. 已知某图的邻接表为



(1). 写出此邻接表对应的邻接矩阵；

(2). 写出由 v1 开始的深度优先遍历的序列；

(3). 写出由 v1 开始的深度优先的生成树；

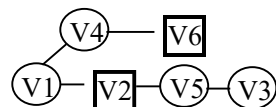
(4). 写出由 v1 开始的广度优先遍历的序列；

(5). 写出由 v1 开始的广度优先的生成树。

答：(1) 略

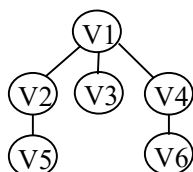
(2) V1V2V5V3V4V6

(3)



(4) V1V2V3V4V5V6

(5)



4. 简述 TCP 和 UDP 协议的主要特点和应用场合。

答：UDP 的主要特点是：

- (1) 传送数据前无需建立连接，没有流量控制机制，数据到达后也无需确认。
- (2) 不可靠交付，只有有限的差错控制机制。
- (3) 报文头部短，传输开销小，时延较短。因此，UDP 协议简单，在一些特定的应用中运行效率高。通常用于可靠性较高的网络环境（如局域网）或不要求可靠传输的场合，另外也常用于客户机 / 服务器模式中。

TCP 的主要特点是：

- (1) 面向连接，提供了可靠的建立连接和拆除连接的方法，还提供了流量控制和拥塞控制的机制。
- (2) 可靠交付，提供了对报文段的检错、确认、重传和排序等功能。
- (3) 报文段头部长，传输开销大。因此，TCP 常用于不可靠的互联网中为应用程序提供面向连接的、可靠的、端到端的字节流服务。

5. 设某路由器建立如下路由表：

目的网络	子网掩码	下一条路由器
128. 96. 39. 0	255. 255. 255. 128	接口 0
128. 96. 39. 128	255. 255. 255. 128	接口 1
128. 96. 40. 0	255. 255. 255. 128	R2
192. 4. 153. 0	255. 255. 255. 196	R3
*（默认）	— —	R4

(1) 现收到 5 个分组，其目的 IP 地址分别为：

- ①128. 96. 39. 10, ②128. 96. 40. 20, ③128. 96. 40. 153, ④ 192. 4. 153. 12,
- ⑤192. 4. 153. 90。

试分别计算其下一跳。

(2) 路由协议的作用是什么？

(3) 在什么情况下要选择多协议路由器？

答：(1)

目的 IP 地址 128. 96. 39. 10 与子网掩码 255. 255. 255. 128 相与得 128. 96. 39. 0，可见该分组经接口 0 转发。

目的 IP 地址 128. 96. 40. 20 与子网掩码 255. 255. 255. 128 相与得 128. 96. 40. 0. 不等于 128. 96. 39. 0。与子网掩码 255. 255. 255. 128 相与得 128. 96. 40. 0，经查路由表可知，该项分组经 R2 转发。

目的 IP 地址 128.96.40.151 与子网掩码 255.255.255.128 相与后得 128.96.40.128，与子网掩码 255.255.255.192 相与后得 128.96.40.128，经查路由表知，该分组转发选择默认路由，经 R4 转发。

目的 IP 地址 192.4.153.17 与子网掩码 255.255.255.128 相与后得 192.4.153.0。与子网掩码 255.255.255.192 相与后得 192.4.153.0，经查路由表知，该分组经 R3 转发。

目的 IP 地址 192.4.153.90 与子网掩码 255.255.255.128 相与后得 192.4.153.0。与子网掩码 255.255.255.192 相与后得 192.4.153.64，经查路由表知，该分组转发选择默认路由，经 R4 转发。

(2) 路由协议用于路由器之间不断地交换路由信息，并根据接到的信息，运行路由算法，优化更新路由，维持路由器有一个动态的优化的路由表。

(3) 路由器可以通过不同类型的网卡分别连接不同类型的局域网。如果互联的局域网高层采用了不同协议，这时就需要使用多协议路由器。

五. 算法设计 (20 分)

(请使用类 C 语言进行编程，如果编码困难可以写伪代码，会适当扣分)

设计算法：二叉树结点的平衡因子 (bf) 定义为该结点的左子树高度与右子树高度之差。

树的结构如下图所示 编写递归算法计算二叉树中各个结点的平衡因子。

(20 分)

```
typedef struct BiTree
```

```
{ int data;
```

```
int bf; //平衡因子
```

```
struct BiTree *rchild;
```

```
struct BiTree *lchild;
```

```
} * BiTree;
```

[题目分析] 由定义，结点的平衡因子 bf 等于结点的左子树高度与右子树高度之差，设计一遍历算法，在遍历结点时，求结点的左子树和右子树的高度，然后得到结点的平衡因子。

```
int Height(BiTree bt)//求二叉树 bt 的深度
```

```
{int hl,hr; if (bt==null)
```

```
return(0);
```

```
else {hl=Height(bt->lchild); hr=Height(bt->rchild); if(hl>hr)
```

```
return (hl+1); else return(hr+1);
```

```
} } // Height
```

```
void
```

```
Balance(BiTree bt)
```

```
//计算二叉树 bt 各结点的平衡因子
```

```
{if (bt)
```

```
{Balance(bt->lchild); //后序遍历左子树 Balance(bt->rchild); //后
```

```
序遍历右子树 hl=Height(bt->lchild); hr=Height(bt->rchild); //求左
```

```
右子树的高度
```

```
    bt->bf=hl-hr; //结点的平衡因子 bf }  
} //算法结束
```