

2018 年华中科技大学 834 计算机专业基础综合

复习八套卷 二 (青蛙版)

参考答案

一. 填空题 (20 分, 每题 2 分)

1. 广义表(a, (a, b), d, e, ((i, j), k)) 的长度是 5, 深度是 3。
2. 循环队列用数组 A[0..m-1] 存放其元素值, 已知其头尾指针分别是 front 和 rear, 则当前队列的元素个数是 (rear-front+m) % m。
3. 对于一个具有 n 个结点的单链表, 在已知的结点*p 后插入一个新结点的时间复杂度为 O(1), 在给定值为 x 的结点后插入一个新结点的时间复杂度为 O(n)。
4. 已知数组 A[0..9, 0..9] 的每个元素占 5 个存储单元, 将其按行优先次序存储在起始地址为 1000 的连续的内存单元中, 则元素 A[6, 8] 的地址为 1340。
5. 高度为 K 的完全二叉树至少有 2^{k-2} 个叶子结点。
6. 如果含 n 个顶点的图形形成一个环, 则它有 n 棵生成树。
7. IP 使用 D 类地址支持多播, 其地址范围是 224.0.0.0~239.255.255.255。
8. 目前常用的四种信道复用方式是: 频分复用、时分复用、码分复用和波分复用。
9. 在路由表中, 对第一条路由最主要的是 目的网络地址 和 下一跳地址。
10. 一个 TCP 报文段分为首部和 数据 两部分, TCP 首部的最小长度是 20 字节。

二. 判断题 (20 分, 每个 2 分)

No.	1	2	3	4	5	6	7	8	9	10
Answer	✗	✗	✓	✓	✓	✗	✗	✗	✓	✗

1. 数组是同类型值的集合。 (✗)
2. 取线性表的第 i 个元素的时间同 i 的大小有关。 (✗)
3. 若输入序列为 1, 2, 3, 4, 5, 6, 则通过一个栈可以输出序列 3, 2, 5, 6, 4, 1. (✓)
4. 一个广义表可以为其它广义表所共享。 (✓)
5. 最小生成树的 KRUSKAL 算法是一种贪心法 (GREEDY)。 (✓)
6. 当改变网上某一关键路径上任一关键活动后, 必将产生不同的关键路径。 (✗)
7. TCP 适用于组播和广播通信方式, 而 UDP 只适用于单播。 (✗)

8. OSI 参考模型中的数据链路层的主要功能是负责分组流控制、差错控制等。
(×)

9. 应用层的网络应用程序分为直接网络应用和间接网络应用两类。(√)

10. RIP 是一种距离矢量路由选择算法，现在仍然广泛使用，适合于大型网络。
(×)

三. 选择题 (30 分, 每个 3 分)

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

1. 设 n 是描述问题规模的非负整数，下面程序片段的时间复杂度是 (A)。

```
x=2; while(x<n/2)  
    x=2*x+1;
```

- A. $O(\log_2 n)$ B. $O(n)$
C. $O(n \log_2 n)$ D. $O(n^2)$

2. 以下数据结构中，(A) 是非线性数据结构

- A. 树 B. 字符串 C. 队 D. 栈

3. 由 3 个结点可以构造出多少种不同的二叉树？ (D)

- A. 2 B. 3 C. 4 D. 5

4. 设哈希表长为 14，哈希函数是 $H(key) = key \% 11$ ，表中已有数据的关键字为 15, 38, 61, 84 共四个，现要将关键字为 49 的结点加到表中，用二次探测再散列法解决冲突，则放入的位置是 (D)

- A. 8 B. 3 C. 5 D. 9

5. 用直接插入排序方法对下面四个序列进行排序（由小到大），元素比较次数最少的是
(C)。

- A. 94, 32, 40, 90, 80, 46, 21, 69 B. 32, 40, 21, 46, 69, 94, 90, 80
C. 21, 32, 46, 40, 80, 69, 90, 94 D. 90, 69, 80, 46, 21, 32, 94, 40

6. 在有向图 G 的拓扑序列中，若顶点 V_i 在顶点 V_j 之前，则下列情形不可能出现的是
(D)。

- A. G 中有弧 $\langle V_i, V_j \rangle$ B. G 中有一条从 V_i 到 V_j 的路径
C. G 中没有弧 $\langle V_i, V_j \rangle$ D. G 中有一条从 V_j 到 V_i 的路径

7. 以下没有采用存储转发技术的交换方式是 (A)。

- A. 电路交换
B. 报文交换
C. 分组交换
D. 信元交换

【解析】 “电路交换”又称为“线路交换”，是一种面向连接的服务。电路交换没有采用存储转发机制，而报文交换和分组交换则采用了存储转发机制。

8. 在以太网中，当一台主机发送数据时，总线上所有计算机都能检测到这个数据信号，只有数据帧中的目的地址与某主机的地址一致时，该主机才接收这个数据帧。这里所提到的地址是（ A ）。

- A. MAC 地址
- B. IP 地址
- C. 端口
- D. 地理位置

【解析】 数据帧中的源地址和目的地址指的是 MAC 地址。

9. 局域网的协议结构一般不包括（ A ）。

- A. 网络层
- B. 数据链路层
- C. 物理层
- D. 媒体访问控制层

【解析】 局域网中的所有主机都处于同一个网段中，不需要路由器或更上层设备将数据转发到不同网段，在局域网中只需要物理地址就可以实现主机间的通信，物理地址属于数据链路层的地址，因此局域网仅涉及数据链路层和物理层，不会包括网络层及其上层。

10. 考虑在一条 1000 米长的电缆（无中继器）上建立一个 $1Gb/s$ 速率的 CSMA / CD 网络，假定信号在电缆中的速度为 $2 * 10^8$ 米 / 秒。最小帧长是（ D ）。

- A. 1220 字节
- B. 1230 字节
- C. 1280 字节
- D. 1250 字节

【解析】 本题考查最小帧长的计算，信道中的往返传播延时 $= 2 * 1000 / (2 * 10^8) = 10\mu s = 10 - 5s$ 。在 $1Gb/s$ 即 $10^9 b/s$ 的速率下，所以最小帧长 = 数据发送速率 * 往返传播延时 $= 10^9 * 10 - 5 = 10000b = 1250$ 字节。

四. 简答题 (60 分)

1. 某算法的计算时间为： $T(n) = 4T(n/2) + O(n)$ ，其中 $T(1) = O(1)$ ，求其时间复杂度，写出具体过程。

答：

假设有递推关系式 $T(n) = aT\left(\frac{n}{b}\right) + f(n)$ ，其中 n 为问题规模， a 为递推的子问题数量， $\frac{n}{b}$ 为每个子问题的规模（假设每个子问题的规模基本一样）， $f(n)$ 为递推以外进行的计算工作。

$a \geq 1, b > 1$ 为常数， $f(n)$ 为函数， $T(n)$ 为非负整数。则有以下结果：

(1) 若 $f(n) = O(n^{\log_b a - \epsilon})$, $\epsilon > 0$, 那么 $T(n) = \Theta(n^{\log_b a})$

(2) 若 $f(n) = \Theta(n^{\log_b a})$, 那么 $T(n) = \Theta(n^{\log_b a} \log n)$

(3) 若 $f(n) = \Omega(n^{\log_b a + \epsilon})$, $\epsilon > 0$, 且对于某个常数 $c < 1$ 和所有充分大的 n 有 $af\left(\frac{n}{b}\right) \leq cf(n)$, 那么 $T(n) = \Theta(f(n))$.

设 $T(n) = 4T(n/2) + n$,

则 $a = 4$, $b = 2$, $f(n) = n$, 代入得 $T(n) = O(n^2)$ 。

2. 采用哈希函数 $H(k) = 3*k \bmod 13$ 并用线性探测开放地址法处理冲突，在数列地址空间 $[0..12]$ 中对关键字序列 $22, 41, 53, 46, 30, 13, 1, 67, 51$ 。

- (1) 构造哈希表（画示意图）；
- (2) 装填因子；
- (3) 成功的平均查找长度。
- (4) 不成功的平均查找长度。

答：

(1)

散列地址	0	1	2	3	4	5	6	7	8	9	10	11	12
关键字	13	22		53	1		41	67	46		51		30
比较次数	1	1		1	2		1	2	1		1		1

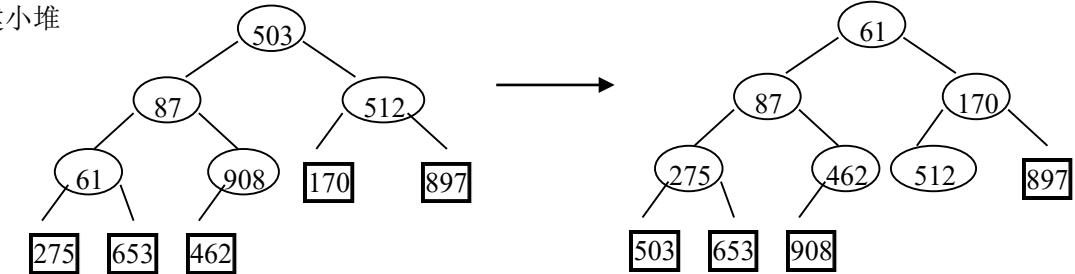
(2) 装填因子 $= 9/13 = 0.7$ (3) $ASL_{succ} = 11/9$ (4) $ASL_{unsucc} = 29/13$

3. 已知待排序的序列为 $(503, 87, 512, 61, 908, 170, 897, 275, 653, 462)$, 试完成下列各题。(12分)

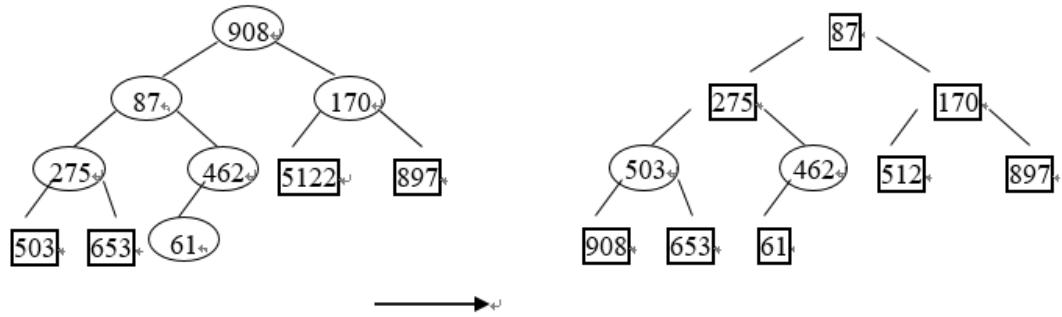
- (1) 根据以上序列建立一个堆（画出第一步和最后堆的结果图），希望先输出最小值。
- (2) 输出最小值后，如何得到次小值。（并画出相应结果图）

答：

(1) 建小堆



(2) 求次小值

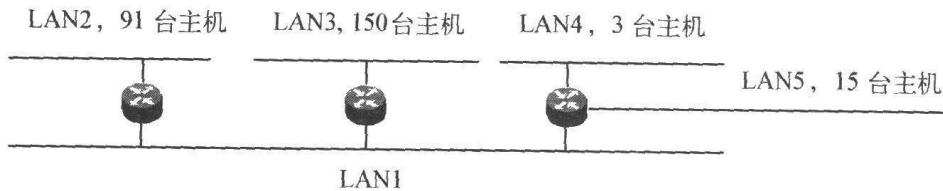


4. 欲构建一个数据传输率为 $1\text{Gb} / \text{s}$ 的千兆以太网，假设电缆长度为 1km ，其中无中继器，信号在电缆中的传播速度为 $2 * 10^8\text{m} / \text{s}$ 。则帧的最小长度是多少？

答：已知电缆的长度为 1km ，信号在电缆中的传播速度为 $2 * 10^8\text{m} / \text{s}$ ，则信号的单向传播时延为 $1 / 200000\text{s}$ ，

往返时延 $= 2 * 1 / 200000 = 1 / 100000\text{s}$ ，即为该网络的争用期。为了按照 CSMA / CD 的方式，最小帧的发送时间不能小于 $1 / 100000\text{s}$ ，否则会被当成无效帧。以 $1\text{Gb} / \text{s}$ 的数据传输速率发送数据， $1 / 100000\text{s}$ 内可发送的比特数为 $109 * 1 / 100000 = 10\text{kb}$ 。因此，帧的最小长度为 10kb 。

5. 一个自治系统有 5 个局域网，如下图所示，LAN2 至 LAN5 上的主机数分别为：91、150、3 和 15，该自治系统分配到的 IP 地址块为 $30.138.118 / 23$ ，试给出每一个局域网的地址块(包括前缀)。



答：分配网络前缀应先分配地址数较多的前缀。已知该自治系统分配到的 IP 地址块为 $30.138.118 / 23$ 。

LAN3：主机数 150，由于 $(2^7 - 2) < 150 + 1 < (2^8 - 2)$ ，所以主机号为 8bit，网络前缀为 24bit。取第 24 位为

0，分配地址块 $30.138.118.0 / 24$ 。

LAN2：主机数 91，由于 $(2^6 - 2) < 91 + 1 < (2^7 - 2)$ ，所以主机号为 7 bit，网络前缀为 25bit。取第 24, 25 位

10，分配地址块 $30.138.119.0 / 25$ 。

LAN5：主机数为 15，由于 $(2^4 - 2) < 15 + 1 < (2^5 - 2)$ ，所以主机号为 5 bit，网络前缀 27bit。取第 24, 25,

26, 27 位为 1110，分配的地址块为 $30.138.119.192 / 27$ 。

LAN1：共有 3 个路由器，再加上一个网关地址，至少需要 4 个 IP 地址。由于 $(2^2 - 2) < 3 + 1 < (2^3 - 2)$ ，所以主机号为 3 bit，网络前缀 29bit。取第 24, 25, 26, 27, 28, 29 位为 111101，分配的地址块为 $30.138.119.232 / 29$ 。

LAN4：主机数为 3，由于 $(2^2 - 2) < 3 + 1 < (2^3 - 2)$ ，所以主机号为 3 bit，网络前缀 29bit。取第 24, 25, 26, 27, 28, 29 位为 111110，分配的地址块为 $30.138.119.240 / 29$ 。

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)

设计非递归算法求树的深度。(20 分)

```
typedef struct BiTree {  
    int data;  
    struct BiTree *rchild;  
    struct BiTree *lchild;  
} * BiTree;
```

[题目分析]由孩子兄弟链表表示的树, 求高度的递归模型是: 若树为空, 高度为零; 若第一子女为空, 高度为 1 和兄弟子树的高度的大者; 否则, 高度为第一子女树高度加 1 和兄弟子树高度的大者。其非递归算法使用队列, 逐层遍历树, 取得树的高度。

```
int height(CSTree t)      //非递归遍历求以孩子兄弟链表表示的树的高度  
{if(t==null)  return(0); else{int front=1, rear=1;  
//front, rear 是队头队尾元素的指针  
  
    int last=1, h=0;      //last 指向树中同层结点中最后一个结点, h 是树的高度  
    Q[rear]=t;            //Q 是以树中结点为元素的队列  
    while(front<=last)  
        {t=Q[front++];      //队头出列  
         while(t!=null)      //层次遍历  
             {if (t->firstchild) Q[++rear]=t->firstchild; //第一子女入队  
              t=t->nextsibling; //同层兄弟指针后移  
             }  
         if(front>last)      //本层结束, 深度加 1 (初始深度为 0)  
             {h++;last=rear;} //last 再移到指向当前层最右一个结点  
        } //while(front<=last)  
    } //else  
} //Height
```