

教务处试卷编号:

备注: 试卷背面为演草区 (不准用自带草纸)

装

订

线

课程编号: 13005710

考核方式: 闭卷

考核时间:

主考教师允许携带的用品: 笔、修正液、尺

## 大连海事大学 2011--2012 学年第一学期《数据结构》 试卷 (B)

题号	一	二	三	四	五	六	七	八	九	十	卷面分	平时分	实验分	总分
得分														

## 一、判断 (用“×”或“√”标识“错”与“对”, 每小题 0.5 分, 共 10 分)

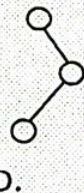
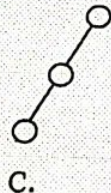
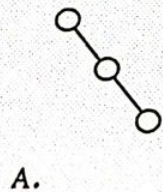
- [ ] 1. 在非空哈夫曼树中, 所有结点的度之和必然是偶数。
- [ ] 2. 若有向无环图中只含一个入度为 0 的顶点, 则该图的顶点拓扑排序序列是唯一的。
- [ ] 3. 在非空完全二叉树中, 若某结点不存在左孩子, 则该结点一定是叶子结点。
- [ ] 4. 给定一棵树的后序遍历序列和先序遍历序列, 则可以唯一地确定这棵树的结构。
- [ ] 5. 哈夫曼树是最优二叉树, 所以根结点到权值最小的叶结点间的路径长度最短。
- [ ] 6. 在 AOE 网中, 若延长某关键路径上的某关键活动的持续时间, 则整个工程的工期将会延期。
- [ ] 7. 具有  $n(n>1)$  个结点的二叉排序树的深度一定小于  $n$ 。
- [ ] 8. 任何基于关键字比较的排序算法的空间复杂度都不大于  $O(n)$ 。
- [ ] 9. 在一棵平衡二叉树上插入新结点  $x$ , 由于做了平衡旋转处理, 所以, 结点  $x$  一定不可能是叶子结点。
- [ ] 10. B\_树是一种多路平衡查找树。
- [ ] 11. 采用孩子-兄弟表示法可以唯一地表示一棵非空树。
- [ ] 12. 对任何长度为  $n>0$  的序列, 任何基于关键字比较的排序算法的时间复杂度都不小于  $O(n \times \log_2 n)$ 。
- [ ] 13. 健壮算法是指, 对于非法的输入, 算法也能给出恰当的反映, 而不会出现莫名其妙的错误。
- [ ] 14. 在非空完全二叉树上, 不存在度为 1 的结点。
- [ ] 15. 因为最小生成树的代价是最小的, 所以每一个连通带权网的最小生成树都是唯一的。
- [ ] 16. 因为具有  $n(n>1)$  个结点的判定树与具有  $n$  个结点的完全二叉树的高度相同, 所以, 度为 2 的结点总数也一定相同。
- [ ] 17. 算法的一个重要特性是至少有 1 个或多个输入。
- [ ] 18. 若  $(u, v)$  是连通网络的一条最小权值的边, 则一定存在一棵最小生成树包含  $(u, v)$ 。
- [ ] 19. 在单链表插入/删除数据元素时无需移动其它元素, 所以, 在单链表插入第  $i$  个元素的操作的时间复杂度为  $O(1)$ 。
- [ ] 20. 针对广义表  $L$  的取表头运算  $HEAD(L)$  的运算结果一定是单元素。





二、选择填空题 (在给定的答案中选择一个正确的答案填在\_\_\_\_处, 每小题 1 分, 共 10 分)

1. 设循环队列存放在一维数组  $Sq.data[0..m]$  中, 则插入数据元素时的队尾指针  $sq.rear$  的操作可表示为\_\_\_\_。  
 A.  $sq.rear = (sq.rear + 1) \text{ MOD } (m + 1)$   
 B.  $sq.rear = (sq.rear + 1) \text{ MOD } (m - 1)$   
 C.  $sq.rear = (sq.front + 1) \text{ MOD } (m + 1)$   
 D.  $sq.rear = (sq.rear + 1) \text{ MOD } (m * 2)$
2. B\_树是一种平衡多路查找树,  $m$  阶 B\_树的定义满足\_\_\_\_。  
 A. 树中每一个非终端结点至多有  $m-1$  个关键字信息  
 B. 根结点至少 2 个指向其子树的指针  
 C. 树中每一个非终端结点至多有  $m$  个指向其子树的指针  
 D. 包括 A, B, C
3. 任一连通图的最小生成树\_\_\_\_。  
 A. 必然有多棵  
 B. 有一棵或多棵  
 C. 只有一棵  
 D. 可能不存在
4. 在最好情况下, 下列算法中\_\_\_\_排序算法所需比较关键字次数最少。  
 A. 简单选择  
 B. 直接插入  
 C. 快速  
 D. 归并
5. 设四维数组  $B[1..3, 2..8, 0..5, 1..8]$  以行主序顺序方法存储在一个连续的存储空间内, 每一个数据元素占 2 个存储单元, 且  $B[1, 4, 0, 1]$  的存储地址是 2000,  $B[2, 3, 4, 5]$  的存储地址是\_\_\_\_。  
 A. 2648  
 B. 2650  
 C. 2652  
 D. A、B、C 全不对
6. 具有 12 个结点的完全二叉树有\_\_\_\_。  
 A. 6 个度为 1 的结点  
 B. 6 个叶结点  
 C. 6 个度为 2 的结点  
 D. 没有度为 1 的结点
7. 假设森林  $F$  中有四棵树, 每棵树的结点个数分别为  $n_1$ 、 $n_2$ 、 $n_3$  和  $n_4$ , 则与森林  $F$  对应的二叉树的根结点的右子树上共有\_\_\_\_个结点。  
 A.  $n_1$   
 B.  $n_1 + n_2 + n_3$   
 C.  $n_1 + n_3 + n_4$   
 D.  $n_2 + n_3 + n_4$
8. 对  $n$  个不同的关键字序列进行冒泡排序, 在\_\_\_\_的情况下比较的次数最多。  
 A. 正序  
 B. 逆序  
 C. 元素无序  
 D. 元素基本有序
9. 折半查找有序表(4, 6, 10, 20, 30, 50, 70, 88, 99), 若查找元素 58, 则它将依次与表中\_\_\_\_元素比较大小, 查找结果是失败。  
 A. 20, 70, 30, 50  
 B. 20, 50, 70  
 C. 30, 70, 50  
 D. 30, 88, 50, 70
10. 设  $T$  是具有 3 个结点的二叉树, 且  $T$  的后序序列与中序序列相同, 则  $T$  的形态为\_\_\_\_。





教务处试卷编号:

专业班级:

学号:

姓名:

备注: 试卷背面为演草区 (不准用自带草纸)

装

订

线

三、给定进栈元素顺序 A、B、C、D、E、F、G, 请给出以 B、A 开头且 F、G 为最后出栈元素的所有可能的出栈序列。 (5 分)

四、请写出递归算法 Checking(11, 7) 执行过程中的所有输出。 (5 分)

```
int Checking (int m, int n)
{
    int k;
    if (m <= n) {
        k = m * n;
    } else {
        k = Checking (m - 2, n - 1) + n;
    }
    printf(k);
    return(k);
} // Checking
```

五、已知关键字序列为 {47, 57, 87, 37, 77, 27, 07, 97, 67, 17, 77}, 按如下要求完成本题。 (6 分)

- (1) 针对给定关键字序列的不同排列, 所构造出的不同形态的二叉排序树中, 在最好和最坏情况下, 该二叉排序树的高度各是多少?
- (2) 根据给定的关键字序列, 构造一棵平衡二叉排序树。
- (3) 在等概率的情况下, 计算查找成功时该平衡二叉排序树的 ASL。

六、选择哈希函数  $H(\text{Key}) = \text{Key} \bmod 13$ , 用开放定址法处理冲突 ( $d_i = 1, 3, 5, 7, 9 \dots$ )。试在 S.data[0..16] 的存储空间上构造下列给定关键字序列的哈希表, 并计算其查找成功时的平均查找长度。 (6 分)

40, 03, 47, 22, 61, 78, 27, 15, 01, 55, 69, 31

七、设主串  $S = \text{'aacbcaacbcaaacbcaacbcdaaacbaaacbcaacbcdaa'}$ , 子串  $T = \text{'aacbcaacbcd'}$ , 求解下列问题。 (6 分)

- (1) 求出模式 T 的 Next[j] 值;
- (2) 求出模式 T 的 NextVal[j] 值;
- (3) 请给出详细的匹配过程, 并回答在 S 中查找 T 至少需要几趟匹配? 至少需要几次比较?





八、将下列给定的关键字序列调整成一个堆，使其满足  $K \geq K_{2i}$  及  $K \geq K_{2i+1}$ ，并依次画出输出前三个较大关键字的过程中所调整成的每一个堆结构。  
03, 31, 78, 40, 05, 61, 93, 11, 47, 17, 27, 71

### 九、证明题 (6 分)

证明：若一个具有  $n$  个结点、 $k$  条边的非连通无向图是一个森林( $n > k$ )，则该森林中必有  $n - k$  棵树。

十、设  $L$  为具有  $n$  ( $n > 0$ ,  $n$  很大且未知) 个数据元素的单链表的头结点指针，试采用 C 程序设计语言编写一个程序，完成将单链表中第  $n/2$  个数据元素之数据元素值为偶数的全部结点删除的功能。要求不另行开辟表结点存储空间，算法的时间复杂度不超过  $O(n)$ 。 (10 分)

单链表结点的抽象数据类型描述如下 (请按照该描述写程序)：

```
typedef struct Lnode {  
    int    data;    //数据元素为整数  
    struct Lnode *next;  
}Lnode, *LinkList;
```

算法的思路和格式要求如下：(请套用这个格式，并请给程序段添加下列注释信息！)

void ABC(LinkList &L) // L 是带头结点的单链表的头指针，是本题算法的唯一参数。

```
{  
    LinkList p, q, s;    //约定程序中的指针型变量为 p, q, s, 其它变量自己定义.
```

//所有的代码请写在答题卡上，写在试卷上的无效！

```
}//ABC
```

(试题完)





一、判断（在下面的表中划“×”或“√”，每小题 0.5 分，共 10 分）

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
√	×	√	√	×	√	×	√	×	√	√	√	√	×	×	×	×	√	×	×

二、填空题（在给定的答案中，选择一个正确的答案填在下表中，每小题 1 分，共 10 分）

1	2	3	4	5	6	7	8	9	10
A	D	B	B	A	B	D	B	C	C

三、答案（共 5 分）

B A C D E F G

B A C E D F G

B A D C E F G

B A D E C F G

B A E D C F G

输出结果每错 1 个扣 1 分，最多扣 5 分。

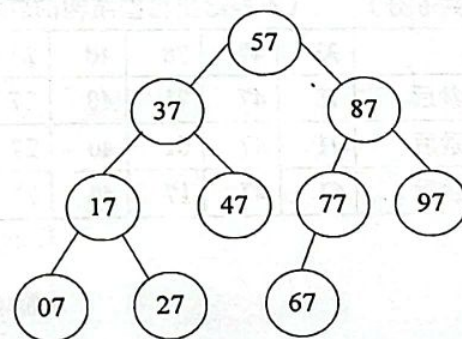
四、答案（共 5 分）

9 13 18 24 31

输出结果每错 1 个扣 1 分，输出顺序颠倒扣 1 分

五、答案（共 6 分）

(1)	最好情况下的二叉排序树的高度为：	4	1 分
	最坏情况下的二叉排序树的高度为：	10	1 分
(2)	在右侧画平衡二叉树		3 分
(3)	$ASL = (3 \times 4 + 4 \times 3 + 2 \times 2 + 1 \times 1) / 10 = 2.9$		1 分



根据结点位置给分





共 5 分

1 分

## 0.5 分

3 分

1分

1 分

1分



九、答案 (共6分)

证明：由题目知，一个具有  $n$  个结点、 $k$  条边的非连通无向图是一个森林( $n > k$ )。

设该森林有  $x$  棵树。这些树的结点数分别为  $n_1, n_2, n_3, \dots, n_x$ 。

因为除了根结点以外，树的每个结点有唯一的分支(即边)与之对应，即结点数=边数+1。

所以这些树的边数分别为  $n_1-1, n_2-1, n_3-1, \dots, n_x-1$ 。

因为  $n = n_1 + n_2 + n_3 + \dots + n_x$

所以  $k = (n_1-1) + (n_2-1) + (n_3-1) + \dots + (n_x-1) = (n_1 + n_2 + n_3 + \dots + n_x) - x = n - x$ ，即  $x = n - k$

故该森林中必有  $n-k$  棵树。

证毕。

给分关键点：结点总数；分支(边)总数；结点与分支(边)间的关系。每个关键点2分。

十、答案 (共10分)

```
typedef struct Lnode {
    int data; //数据元素为整数
    struct Lnode *next;
} Lnode, *LinkList;
```

如果采用先对整个单链表统计数据元素个数的方法，然后再确定第  $n/2$  个数据元素的位置且后面的程序都正确的话，那么可以给5分。

void ABC(LinkList &L) // L 是带头结点的单链表的头指针，是本题算法的唯一参数。

```
{
    LinkList p, q, s;
    p=L; q=L; // (1分)
    while (p)
    { if (p->next) p=p->next->next;
      else {p=p->next; break;}
      q=q->next;
    } // (while 循环 5分)
    s=q; p=q->next; // (1分)
    while (p)
    { if (p->data%2==0) {q=p; s->next=p->next; p=p->next; free(q);}
      else { s=p; p=p->next;}
    } // (while 循环 3分)
}
```

//ABC

代码可以不唯一

