

安徽大学 20_19—20_20 学年第 2 学期

《数据结构》(A 卷) 考试试题参考答案及评分标准

一、算法分析题 (每小题 5 分, 共 25 分)

1. 该算法的时间复杂度为 $O(\sqrt{n})$; 正确给 5 分。

2. 答案与评分标准如下:

(1) 该算法功能是判断链表 head 的长度是偶数还是奇数, 偶数返回 0, 奇数返回 1; (3 分)

(2) 时间复杂度: $O(n)$; 空间复杂度: $O(1)$ (2 分)

3. 答案与评分标准如下:

(1) 该算法功能是从表尾开始依次输出链表的所有结点的值。 (3 分)

(2) 当 $L=\{5,3,1,6,4,2\}$ 时, 执行 Function(L) 后, 输出的结果是: 2 4 6 1 3 5 (2 分)

4. 答案与评分标准如下:

(1) 该算法功能是统计二叉树 root 中叶子结点的数量。 (3 分)

(2) 当 $root=(A(B(C,D),E(F)))$, 执行 Function (root) 后, n 的值为 3。 (2 分)

多少执行 Fun(a,8) 后, 数组 a 的结果是 $a=\{1,3,5,7,2,4,6,8\}$

5. 答案与评分标准如下:

(1) 该算法功能是用增量递减为 $1/3$ 的希尔排序对 a 数组中的元素进行递增排序。 (3 分)

(2) 当 $a[]=\{5,1,3,6,2,7,4,8\}$ 时, 执行 Fun(a,8) 的过程如下:

$d=2: 2,1,3,6,4,7,5,8$ (1 分)

$d=1: 1,2,3,4,5,6,7,8$ (1 分)

二、简答题 (每小题 5 分, 共 15 分)

6. 答案与评分标准如下:

(1) 数组中共有 $6*7*8=336$ 个元素; (2 分)

(2) 元素 $a[3][4][5]$ 的地址为: $1000+(3*7*8+4*8+5)*4=1820$ (3 分)

7. 答案与评分标准如下:

(1) 从 LS 中取出原子 a 的操作为: $H(H(LS))$ (2 分)

(2) 从 LS 中取出原子 d 的操作为: $H(T(H(T(LS))))$ (3 分)

8. 答案与评分标准如下:

(1) 后缀表达式为: $AB*CD+-E+$ (3 分)

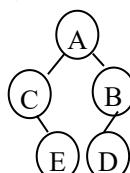
(2) 保存在栈中操作符的最大个数为 3 (2 分)

三、应用题 (9 和 10 题 5 分, 11、12 和 13 题 10 分, 共 40 分)

9. 答案与评分标准如下:

(1) 二叉树如下图所示。(3 分)

(2) 后序遍历序列为: ECDBA (2 分)



10. 答案与评分标准如下:

第一趟快速排序后的关键字序列: (0,2,5,4,3,6,8,7,9,10) (5 分)

正确给出结果给满分。

11. 答案与评分标准如下:

(1) 散列表如下表所示：每正确填入一个关键字给 1 分，全部正确给 5 分：

散列地址	0	1	2	3	4	5	6
关键字	10		7	12	24	34	
比较次数	1		1	2	1	2	

(2) 等概率情况下，查找成功和查找不成功的时平均查找长度分别如下：

查找成功：ASL=1/5 (1+1+2+1+2)=7/5 (3 分)

查找不成功：ASL=1/5(2+1+5+4+3)=3 (2 分)

12. 答案与评分标准如下：

串 T 的 next 和 nextval 值如下表所示，每正确给出一组 next 和 nextval 值给 1 分。

j	1	2	3	4	5	6	7	8	9	10
模式串 T	b	a	c	b	a	b	a	b	a	b
next[j]	0	1	1	1	2	3	2	3	2	3
nextval[j]	0	1	1	0	1	3	1	3	1	3

13. 答案与评分标准如下：

求解的最小生成树如下：

第 1 条边为 (0, 1)，权值为 3；

第 2 条边为 (1, 2)，权值为 2；

第 3 条边为 (2, 3)，权值为 4；

第 4 条边为 (2, 6)，权值为 5；

第 5 条边为 (6, 4)，权值为 1；

第 6 条边为 (4, 5)，权值为 3；

正确给出每一条边，给满分；每错一条边，扣 2 分，直到扣完为止。

四、算法设计题（每小题 10 分，共 20 分）

14. 该题答案不唯一，正确给出算法思想给 4 分，不完全正确的，可酌情给分。

```
LNode * FindIntersectingNode(LinkList list1, LinkList list2)
```

```
{  
    int L1=0, L2=0, diff=0;  
    LinkList head1=list1->next, head2=list2->next;  
    while(head1 != NULL) {  
        L1++; head1=head1->next;  
    }  
    while(head2 != NULL) {  
        L2++; head2=head2->next;  
    }  
    if(L1<L2) {  
        head1=list2; head2=list1; diff = L2-L1;  
    }  
    else {  
        head1=list1; head2=list2; diff = L1-L2;  
    }  
    for(int i=0;i<diff; i++)  
        head1=head1->next;  
    while(head1 != NULL && head2 != NULL) {
```

```

    if (head1 == head2)
        return head1;
    head1 = head1->next;
    head2 = head2->next;
}
return NULL;
}

```

该算法的时间复杂度为: $O(\max(\text{list1 表长}, \text{list2 表长}))$ 。

14 该题答案不唯一, 正确给出算法思想给 4 分, 不完全正确的, 可酌情给分。

```

int FindMaxValue(BiTree root)
{
    BiTree temp;
    int max = INT_MIN;
    Queue Q;
    InitQueue(Q); // 初始化队列 Q
    EnQueue(Q, root); // 入队操作
    while( !IsEmptyQueue(Q)){ // 队列非空
    {
        DeQueue( Q, temp); // 出队操作
        if ( max < temp->data ) max = temp->data;
        if (temp->lchild) EnQueue(Q, temp->lchild);
        if (temp->rchild) EnQueue(Q, temp->rchild);
    }
    DestroyQueue(Q);
    return max;
}

```