安徽大学 20__20__—20__21__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 20 分）

1. 该算法的时间复杂度为 O(n)； 正确给 5 分。

2. 答案与评分标准如下：

（1）该算法功能是查找链表 L 中间位置结点；（3 分）

（2）当 L={1,3,5,7,9,11,13}时，执行 Function(L)后，ptr2->data=7 （2 分）

3. 答案与评分标准如下：

（1）该算法功能是求解二叉树 root 中度等于 1 的结点的个数。 （3 分）

（2）若 root=(A(B(D,E),C(F,G)))，则执行 Function(root)后，n=0 （2 分）

4. 答案与评分标准如下：

（1）该算法功能是在 R[s]~R[t]序列中查找第 k 小元素的关键字。 （3 分）

（2）若 R={35,40,38,11,13,34,48,75,6,19}，执行 Function (R,0,9,4)后，其值为 19。 （2 分）

二、计算题（每小题 5 分，共 10 分）

5. 答案与评分标准如下：

（1）Tail(L)=((c,d)) （2 分）

Head(Tail(L)) =(c,d) （2 分）

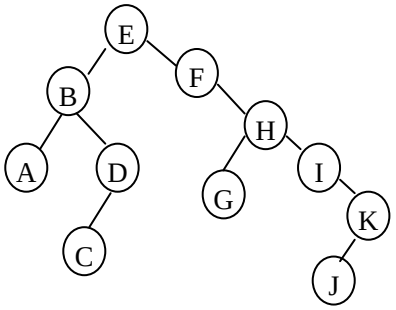
Tail(Head(Tail(L)))=(d)。 （1 分）

6. 答案与评分标准如下：

&a[2,5,6]= &a[0,0,0]+(2*9*10+5*10+6)*4

=1000+(180+50+6)*4 = 1944

地址计算正确给 5 分。



三、应用题（每小题 10 分，共 40 分）

7. 答案与评分标准如下：

二叉树如右图所示。

8.答案与评分标准如下：

(1) [35*,15,35] 45 [60,50,77,58,55,62,98] （2 分）

(2)[15] 35* [35] 45 [60,50,77,58,55,62,98] （2 分）

(3) 15, 35*, 35,45, [55,50,58] 60 [77,62,98] （2 分）

(4) 15, 35*, 35,45, [50] 55 [58] 60 [77,62,98] （2 分）

(5) 15, 35*, 35,45, 50, 55, 58,60, [62] 77 [98] （2 分）

9. 答案与评分标准如下：

（1）散列表如下表所示： （6 分）

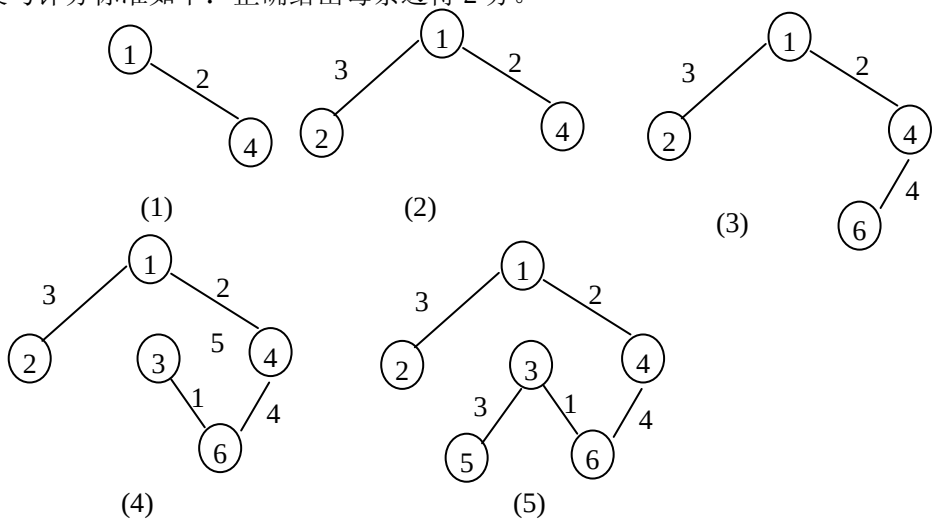
散列地址	0	1	2	3	4	5	6	7	8	9
关键字		8	23	38		12	20	27		
比较次数		1	1	1		1	1	2		

（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=1/6（1*5+2）=7/6 （2 分）

查找不成功：ASL=1/7(1+4+3+2+1+4+3)=18/7 （2 分）

10. 答案与评分标准如下：正确给出每条边得 2 分。



四、算法设计题（每小题 10 分，共 30 分）

11. 该题答案不唯一，不完全正确的，可酌情给分。（10 分）

```
//将带头结点的单链表 List1 和 List2 交替合并成一个带头结点的链表 List1
 LinkList    AlternateMerge((LinkList List1,LinkList List2)
 {
      //p 和 q 为工作指针，初始时分别指向 List1 和 List2 的首元结点
      LNode *p=List1->next,*q=List2->next;
      //temp 为工作指针，指向合并后的链表尾结点，初始时指向 List1
      LNode *temp = List1;
      while(p != NULL && q!=NULL) //依次扫描链表 List1 和 List2
      {
           temp->next=p;
           temp=temp->next;
           p=p->next;
           temp->next=q;
           q=q->next;
           temp=temp->next;
      }
      if(p != NULL)
           temp->next=p;
      else temp->next=q;
      free(List2); //释放 list2 的头结点
      return List1; //返回合并后链表的头指针
 }
```

12．该题答案不唯一，不完全正确的，可酌情给分。

（1）删除一棵二叉树算法如下：（5 分）

```
//后序遍历删除一棵二叉树
 void DelBinaryTree(BiTree root)
 {
```

```
        if(root == NULL)
            return;
        DelBinaryTree(root->lchild);
        DelBinaryTree(root->rchild);
        free(root); //delete(root)
 }
```

（2）求二叉树的高度算法如下：（5 分）

```
int HeightBinaryTree(BiTree root)
{
        int leftheight,rightheight;
        if(root==NULL) return 0;
        else {
            leftheight=HeightBinaryTree(root->lchild);
            rightheight=HeightBinaryTree(root->rchild);
            if(leftheight > rightheight) return (leftheight+1);
            else return (rightheight+1);
        }
}
```

13．该题答案不唯一，不完全正确的，可酌情给分。

```
void InitQueue(LinkQueue &Q)//初始化队列                （2 分）
{
      Q.rear=new QNode();
      Q.rear->next=Q.rear;
}
void EnQueue(LinkQueue &Q, QElemType e) //入队操作      （4 分）
{
      QueuePtr p=new QNode;
      p->data=e;
      p->next=Q.rear->next;
      Q.rear->next=p;
      Q.rear=p;
}
void DeQueue(LinkQueue &Q, QElemType &e) //出队操作      （4 分）
{     QueuePtr p;
      if(Q.rear->next==Q.rear) //队空
          return;
      p=Q.rear->next->next;
      e=p->data;
      if(p==Q.rear) //队列中只有一个元素
          Q.rear=Q.rear->next;
      Q.rear->next=p->next;
      delete p;
}
```