



厦门大学《数据结构》期末试题·答案

考试日期：2006•1 (A)

信息学院自律督导部



- 一、 (本题 15 分) 试设计一个结点数据类型为整型的带表头结点的有序单链表，然后设计一个算法，该算法将这个有序单链表划分成两个单链表，使得第一个单链表中包含原单链表中所有数值为奇数的结点，第二个单链表中包含原单链表中所有数值为偶数的结点，且两个单链表中结点的相对排列顺序与原单链表中相同。注意：要求使用原单链表的空间，表头结点可以另辟空间。

[解答]

```
void split(LinkList &HL, LinkList &L1, LinkList &L2) {
    q1=L1= (LinkList) malloc(sizeof(LNode));
    q2=L2= (LinkList) malloc(sizeof(LNode));
    p=HL->next;
    while (p!=NULL) {
        if (p->date % 2 != 0) {
            q1->next= p; q1=p;}
        else
            q2->next= p; q2=p;}
        p=p->next;
    }
    q1->next=q2->next=NULL;
    free(HL);
}
```

- 二、 (本题 20 分) 试设计一个递归算法，判断二叉树 T 是否是满二叉树，假设 T 是以二叉链表存储。

```
typedef struct BiTNode{
    TElemType data;
    Struct BiTNode *lchild, *rchild;
} BiTNode, *BiTree;
```

解答：

满二叉树中任一个结点为根的子树都是满二叉树。

算法：

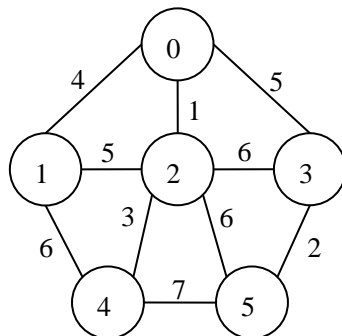
- (1) 如果二叉树 T 是空树，则是满二叉树；
- (2) 如果二叉树 T 非空，左右子树都是满二叉树，而且深度一样，则 T 是满二叉树；
- (3) 如果二叉树 T 非空，左子树或右子树不是满二叉树，则不是满二叉树；

(4) 如果二叉树 T 非空，左右子树都是满二叉树，但深度不一样，则 T 不是满二叉树。

```
//该函数判断二叉树 T 是否是满二叉树
//如果是满二叉树，返回 TRUE，Depth 返回该树的深度；
//否则返回 FALSE，Depth 无定义；
Boolean Check( BiTree T, int &Depth)
{ int ldepth, rdepth;
  if( T==NULL) { Depth=0; return TRUE; }
  if( Check(T->lchild, ldepth)==FALSE ) return FALSE;
  if( Check(T->rchild, rdepth)==FALSE) return FALSE;
  if( ldepth!=rdepth ) return FALSE;
  Depth=ldepth+1; return TRUE;
}
```

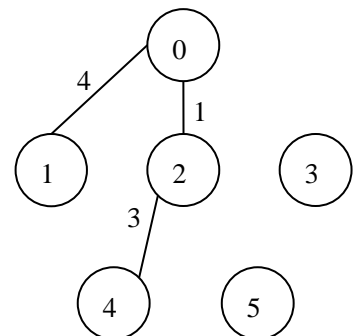
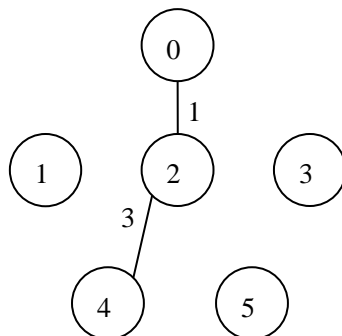
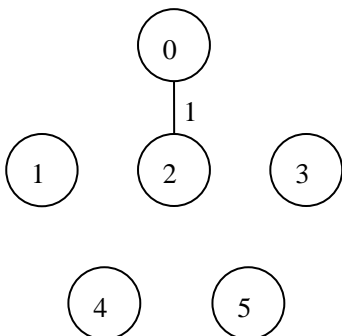
三、 （本题 15 分）给定下面的带权无向图 G：

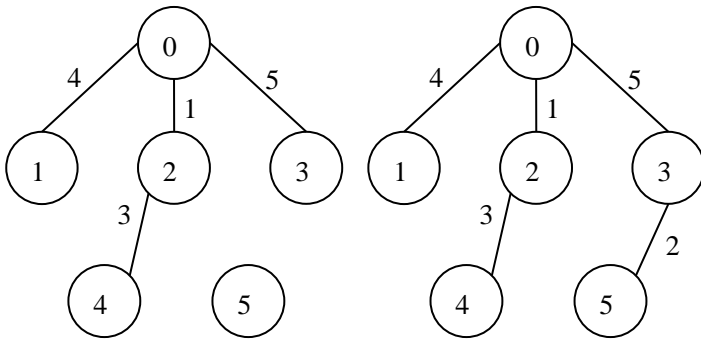
- 1) 从顶点 0 出发，请写出深度优先遍历序列和广度优先遍历序列，当有多种选择时，编号小的结点优先。
- 2) 分别使用普里姆算法和克鲁斯卡尔算法求出下图的最小生成树，仅需画出最小生成树的成长过程即可。



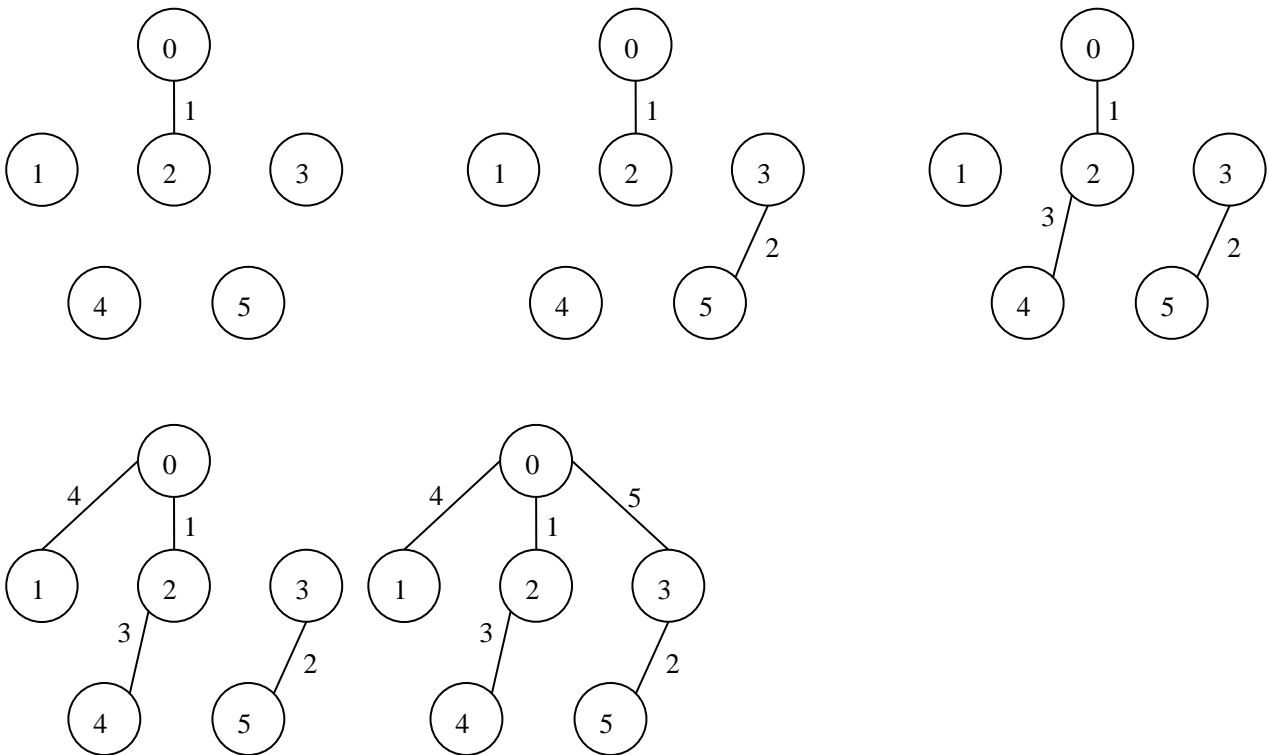
解答：

- (1) 深度优先遍历序列为：0，1，2，3，5，4
广度优先遍历序列为：0，1，2，3，4，5
- (2) 普里姆算法求最小生成树的过程如下：





克鲁斯卡尔算法求最小生成树的过程如下：

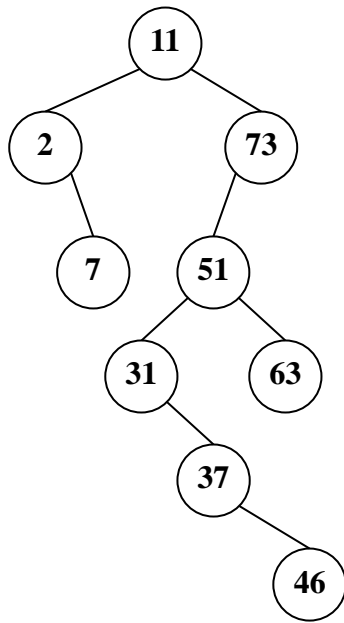


四、（本题 15 分）设有一个关键字序列 {11, 73, 51, 31, 63, 37, 46, 2, 7}，

- 1) 从空树开始构造排序二叉树，画出得到的排序二叉树；分别计算该排序二叉树在等概率下查找成功的平均查找长度和查找失败的平均查找长度；
- 2) 从空树开始构造平衡二叉树，画出每加入一个新结点时二叉树的形态；若发生不平衡，请画出调整平衡后的结果；分别计算该平衡二叉树在等概率下查找成功的平均查找长度和查找失败的平均查找长度。

[解答]

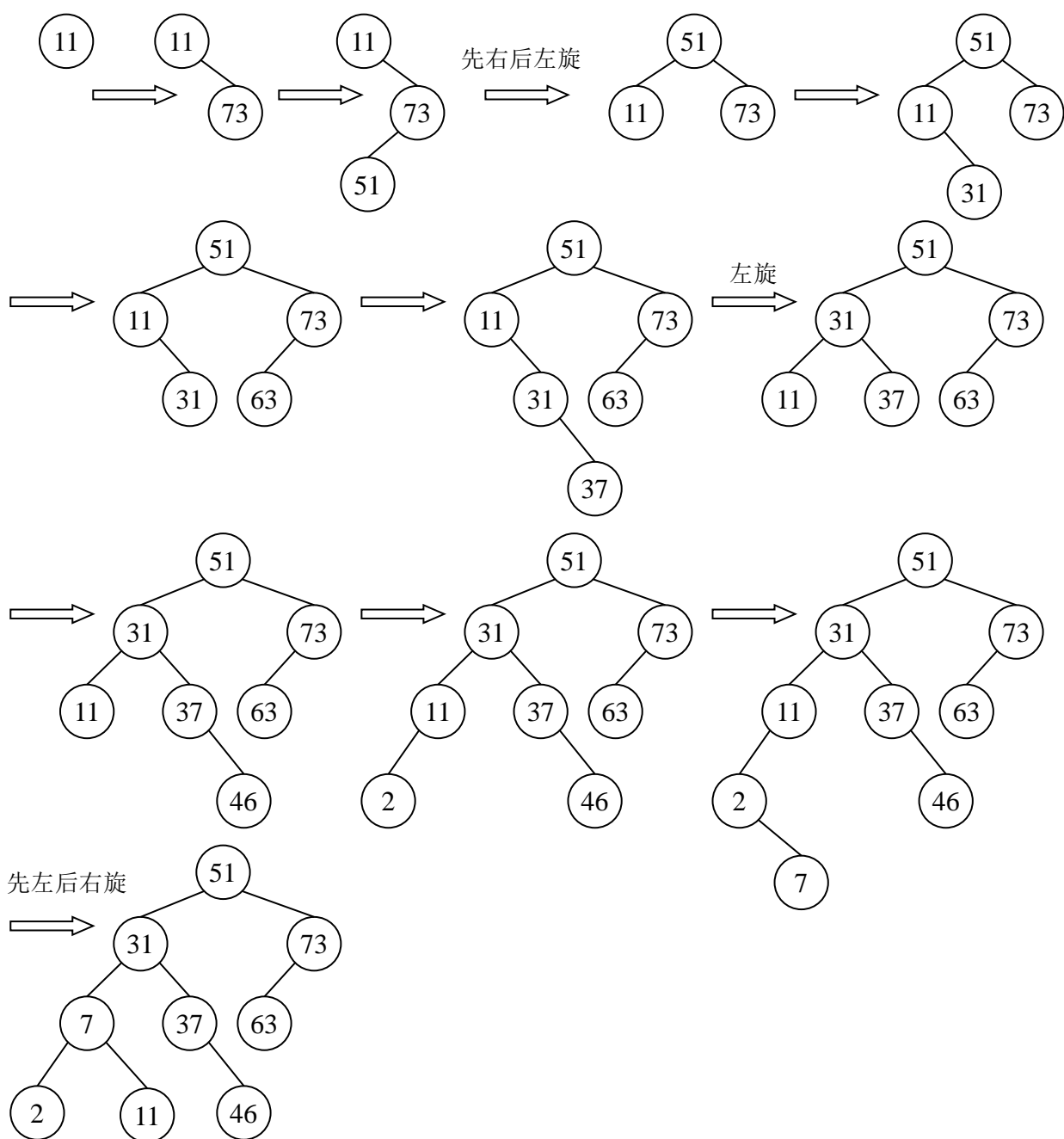
1)



在等概率下，查找成功的平均查找长度 $ASL_{succ} = (1/9) * (1 + 2*2 + 3*2 + 4*2 + 5 + 6) = 10/3$

在等概率下，查找失败的平均查找长度 $ASL_{unsucc} = (1/10) * (2*2 + 3*2 + 4*3 + 5*1 + 6*2) = 3.9$

2)



在等概率下，查找成功的平均查找长度 $ASL_{succ} = (1/9) * (1 + 2*2 + 3*3 + 4*3) = 26/9$

在等概率下，查找失败的平均查找长度 $ASL_{unsucc} = (1/10) * (2*1 + 3*3 + 4*6) = 3.5$

五、（本题 10 分）已知待散列存储的关键字序列为（4, 15, 38, 49, 33, 60, 27, 71），哈希函数为 $H(key) = key \text{ MOD } 11$ ，哈希表 HT 的长度为 11，采用二次探测再散列法解决冲突。试构造此哈希表，并求出在等概率情况下查找成功的平均查找长度。

解：哈希表为

0	1	2	3	4	5	6	7	8	9	10
33	60		27	4	15	38	71		49	

平均查找长度为: $(1+2+2+4+1+5+6+7)/8=3.5$ 。

六、 (本题 15 分) 试设计算法在 $O(n)$ 时间内将数组 $A[1..n]$ 划分为左右两个部分, 使得左边的所有元素为奇数, 右边的所有元素均为偶数, 要求所使用的辅助存储空间大小为 $O(1)$ 。

解: 该题算法的主要思路如下:

(1) 设置两个指针 i 和 j , 其中 $i=1$, $j=n$ 。

(2) 当 $i < j$ 时作如下循环:

i 不断自加从左往右找到第一个偶数

j 不断自减从右往左找到第一个奇数

$A[i]$ 与 $A[j]$ 交换

(3) 算法结束

Adjust(int A[1..n])

```
{
    int i=1, j=n;
    while (i<j){
        while (A[i] % 2 != 0 && i<=n) i++;
        while (A[j] % 2 == 0 && j>=1) j--;
        if (i>n || j<1) break;
        if (i<j) A[i]  $\leftrightarrow$  A[j];
    }
}
```

算法的时间复杂性为 $O(n)$, 辅助存储空间为 $O(1)$ 。

七、 (本题 10 分) 若待排序记录的关键字集合是 $\{30, 90, 27, 4, 48, 15, 9, 13, 18\}$, 欲将其按关键字非递减排序:

1) 若采用快速排序 (选取待排序列中的第一个记录作为枢轴), 请给出第一趟和第二趟排序的结果;

2) 若采用堆排序, 请画出初始建立的“大顶堆”;

3) 当给定的待排序记录的关键字基本有序时, 应采用堆排序还是快速排序? 为什么?

1) 第一趟快速排序的结果: 18 13 27 4 9 15 30 48 90

具体过程: (可省略不写)

30 90 27 4 48 15 9 13 18

18 90 27 4 48 15 9 13 (30)

18 (30) 27 4 48 15 9 13 90

18 13 27 4 48 15 9 (30) 90

18 13 27 4 (30) 15 9 48 90

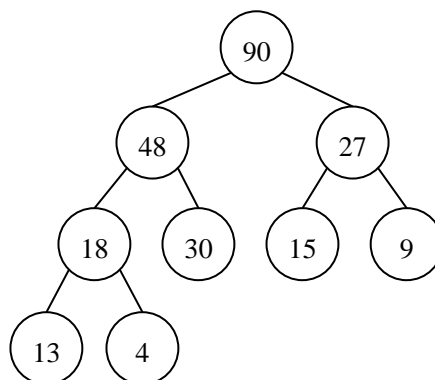
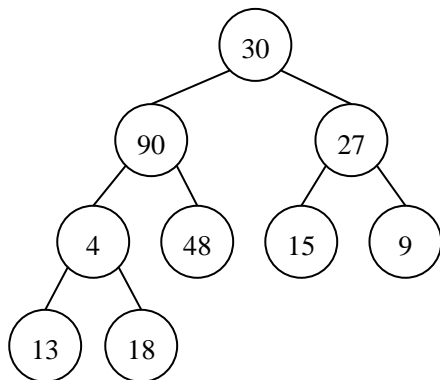
18 13 27 4 9 15 (30) 48 90

第二趟快速排序的结果: 15 13 9 4 18 27 30 48 90

具体过程: (可省略不写)

18 13 27 4 9 15 (30) 48 90
 15 13 27 4 9 (18) (48) 90
 15 13 (18) 4 9 27
 15 13 9 4 (18) 27

2) 调整前的初始情况如左下图，初始大顶堆如右下图：



3) 应采用堆排序，因为当给定的待排序的记录的关键字基本有序时，快速排序将退化为起泡排序，此情况下的快速排序时间复杂度为 $O(n^2)$ ，而堆排序即便在最坏情况下的时间复杂度仍为 $O(n\log n)$ 。

八、（本题不记分）请谈谈学习《数据结构》课程的心得体会，或对该课程的教学提出意见和建议。