2021-2022 学年第一学期《数据结构》期末考试试卷(A 卷)

参考答案及评分标准

一、单项选择题(本大题共 10 小题，每小题 1 分，共 10 分)

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

二、简答题(本大题共 2 小题，每小题 5 分，共 10 分)

1. 在单循环链表中设置尾指针比设置头指针好吗？为什么？

答：在单循环链表中设置尾指针比设置头指针好。----------1 分

尾指针是指向终端结点的指针，用它来表示单循环链表可以使得查找链表的开始结点和终端结点都很方便。----------1 分

设一带头结点的单循环链表，其尾指针为 rear，则开始结点和终端结点的位置分别是 rear->next->next 和 rear, 查找时间都是 O(1)。----------2 分

若用头指针来表示该链表，则查找终端结点的时间为 O(n)。----------1 分

2. 折半查找适不适合链表结构的序列，为什么？用折半查找的查找速度必然比线性查找的速度快，这种说法对吗？

答：不适合！----------1 分

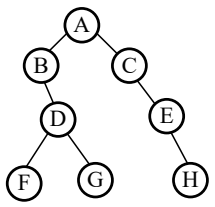
虽然有序单链表的结点是按从小到大（或从大到小）顺序排列，但因其存储结构为单链表，查找结点时只能从头指针开始逐步搜索，故不能进行折半查找。-----2 分

折半查找的速度在一般情况下是快些，但在特殊情况下未必快。----------1 分

例如，所查数据位于首位时，则线性查找快，而二分查找则慢得多。---------1 分

三、综合应用题(本大题共 6 小题，每小题 10 分，共 60 分)

1. 答：

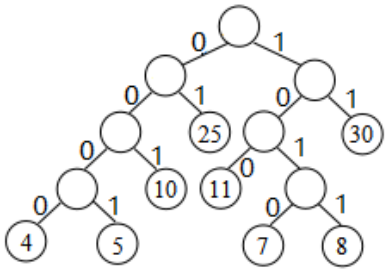


----------6 分

其后序遍历序列为：F、 G、 D、 B、 H、 E、 C、 A ----------4 分

2. 答：根据题意，设这 8 个字母对应的权值分别为(5，25，4，7，10，11，30，8)，并且 n=8。----------1 分

按照左小右大原则构造的哈夫曼树为：

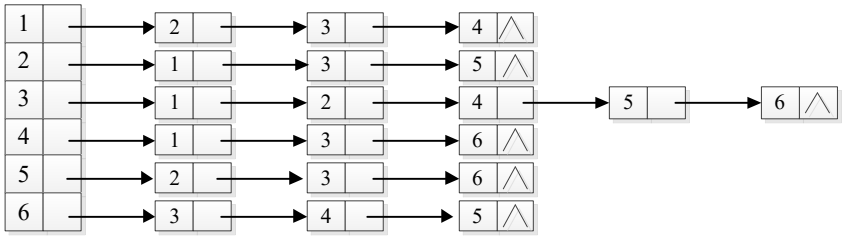


----------4 分

各字符相应哈夫曼编码为：A:0001 B:01 C:0000 D:1010 E:001 F:100 G:11 H:1011 ---4 分

WPL=(4+5+7+8)*4+(10+11)*3+(25+30)*2=269 ----------1 分

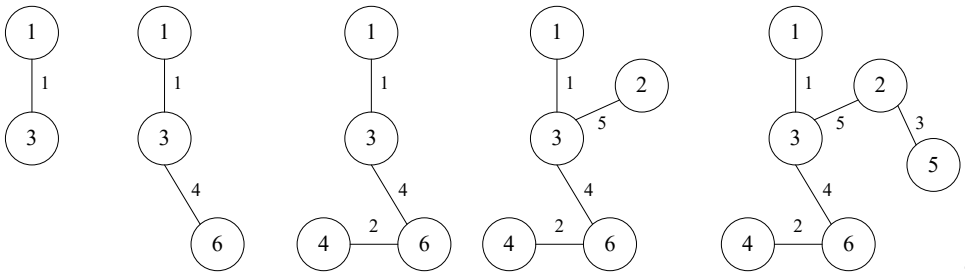
3. 答：(1)：



---------3 分

(2)：1, 2, 3, 4, 5, 6 ----------2 分

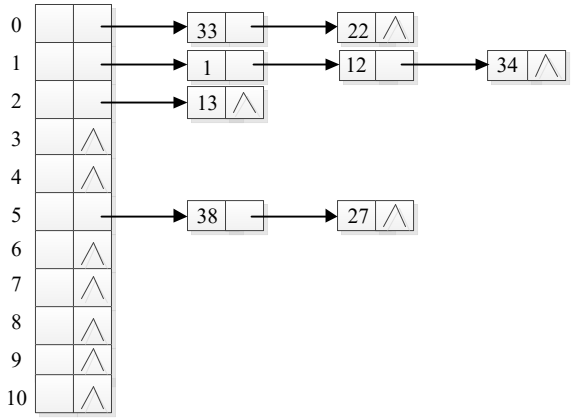
(3)：



----5 分

(注：此最小生成树形态不定，只要包含全部相应边即可！)

4. 答：(1)：

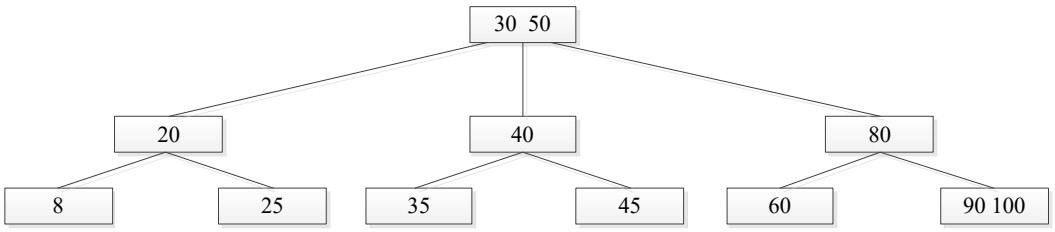


---------6 分

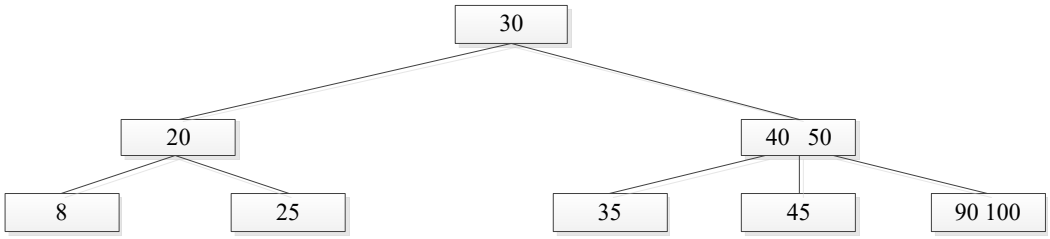
(2)：$\mathbf{ASL}_{succ}$= (1 × 4 + 2× 3 +3×1)/8 = 13/8 ----------2 分

(3)：ASL_{unsucc}= (3 + 4 + 2+1+1+3+1+1+1+1+1)/11 = 19/11 ------2 分

5. 答：(1): ----------6 分



(2): ----------4 分



6. 答：(1) 34 27 38 14 26 46 86 65 53 74 ----------3 分

(2) 26 27 14 34 38 46 74 65 53 86 ----------3 分

(3) 14 26 27 34 38 46 53 65 74 86 ----------3 分

(4) 14 26 27 34 38 46 53 65 74 86 ----------1 分

四、算法设计题(本大题共 2 小题，每小题 10 分，共 20 分)

1. Delete(LinkList head, int min, int max) ----------0.5 分

```
{
  Lnode   *p, *q, *t;  ---------0.5 分
  if (head!=NULL)     ---------0.5 分
  {
    q=head;              ----------0.5 分
    p=head->next;      ----------0.5 分
    while((p!=NULL) && (p->data<=min))//寻找满足值<=min 的最大值结点 q
                                                          ----------1 分
    {
        q=p;            ----------0.5 分
        p=p->next; ----------0.5 分
    }
    while((p!=NULL) && (p->data<max))//寻找满足值>=min 的最小值结点 p
                                                          ----------1 分
      p=p->next;     ----------1 分
    while(q->next!=p)//删除值在(min, max)之间的结点  ----------1 分
```

```
            {
                t=q->next;    ----------0.5 分
                q->next=t->next;    ----------0.5 分
                free(t);    ----------1 分
            }
            q->next=p;    ----------0.5 分
        }
    }
```

2. int LeafCount_BiTree(Bitree T)//求二叉树中叶子结点的数目 --------1 分

```
{     ----------0.5 分
  if(!T) return 0; //空树没有叶子  ---------2 分
  else if(!T->lchild&&!T->rchild) return 1; //叶子结点  ----------2 分
  else return    Leaf_Count(T->lchild)+Leaf_Count(T->rchild);//左子树的叶子数加
  上右子树的叶子数          ----------4 分
} ----------0.5 分
```