一、选择题（20分）

ACDCD DBCAC

二、填空题（20分，每空2分）

1．2i-1，（n+1）/2，（n-1）/2

2．n-1，n+1

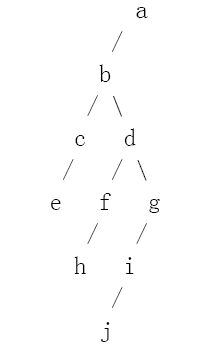
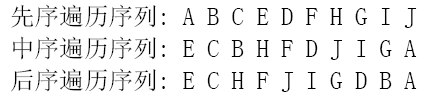
3. 24和53

4. 70

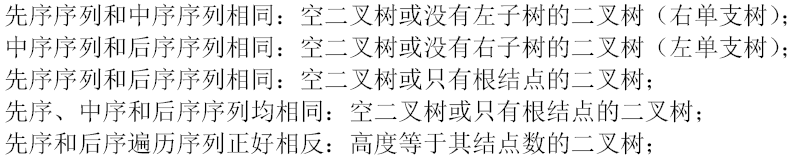
5.一个“右-右”旋和两个“右-左”旋

6.P，H->Elements[i]=H->Elements[i/2]

三、（5分）

四、（5分）



五、（10分）

8个字母的哈夫曼编码：a为1010，b为00，c为10000，d为1001，e为11，f为10001，g为01，h为1011。

如果用3位二进制数对这8个字母进行等长编码，其平均长度为3；

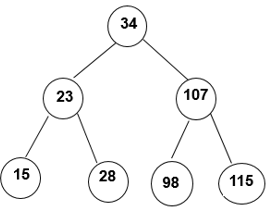
而根据哈夫曼编码的平均码长为：

4\*0.07+2\*0.19+5\*0.02+4\*0.06+2\*0.32+5\*0.03+2\*0.21+4\*0.10=2.61。

因此哈夫曼编码的平均码长是等长编码平均码长的2.61/3=0.87=87%；

它使电文总长平均压缩了13%

六、（10分）



七、(15分)

**BinTree Create\_Bintree( ) {**

**BinTree BT;**

**char ch;**

**if((ch = getchar( )) == '0')**

**BT = NULL;//空子树**

**else{**

**BT = (BinTree)malloc(sizeof(TNode));**

**BT ->Data = ch;**

**BT ->Left=Create\_Bintree( );**

**BT ->Right=Create\_Bintree( ); }**

**return BT; }**

八、(15分)

1. WPL=4\*3+2\*35+3\*13+3\*15+2\*20+4\*5+3\*9=253
2. **// 计算哈夫曼树带权路径长度WPL**

**ElemType calculateWeightLength(HuffmanTree ptrTree,int len){**

**if(ptrTree==NULL){ //空树返回0**

**return 0; }**

**else{**

**if(ptrTree->left==NULL && ptrTree>right==NULL) {**

**return ptrTree->weight \* len; }**

**else{**

**return calculateWeightLength(ptrTree->left,len+1) + calculateWeightLength(ptrTree->right,len+1); //向下递归计算**

**}**

**}**

**}**