赫夫曼（Huffman）树

赫夫曼树，又称最优树，是一类带权路径长度最短的树

路径以及路径长度

从树中的一个结点到另一个结点之间的分支构成这两个结点之间的路径，路径上的分支数目称做路径长度

树的路径长度

树的路径长度是从树根到每一结点的路径长度之和

结点的带权路径长度

结点的带权路径长度为从该结点到树根之间的路径长度与该结点上权的乘积

树的带权路径长度

树的带权路径长度为树中所有叶子结点的带权路径长度之和

最优二叉树

假设有n个权值，构造一颗有n个叶子结点的二叉树，每个叶子结点带权为wi，则其中带权路径长度WPL最小的二叉树称作最优二叉树或赫夫曼树

构造

```
//循环结束条件：1.没有任何结点；2.有结点，最后合并到只有一个结点
    while(!isEmpty(list) && getLength(list) != 1){
        //寻找到两个最小的数值
        void *temp;
        BinaryTree minTree1, minTree2;

        //获取到值最小的结点并从列表中删除
        deleteElement(list,getMinTree(list),temp);
        minTree1 = (BinaryTree)temp;

        //获取到值最第二小的结点并从列表中删除
        deleteElement(list,getMinTree(list),temp);
        minTree2 = (BinaryTree)temp;

        //计算新权值并设置其左右子树
        BinaryTreeElementType newNodeData = minTree1->data + minTree2->data;
        BinaryTree newNode = (BinaryTreeNode *)malloc(sizeof(BinaryTreeNode));
        newNode->data = newNodeData;
        newNode->leftChild = minTree1;
        newNode->rightChild = minTree2;

        //将新树添加到列表中
```

```
        insertElement(list, getLength(list), newNode);
    }
```

赫夫曼编码

若要设计长短不等的编码，必须是任一字符的编码都不是另一个字符的编码的前缀，这种编码叫做前缀编码（无子女）

可以利用二叉树来设计二进制的前缀编码，约定左分支表示字符‘0’，右分支表示字符‘1’

则可以从根结点到叶子结点的路径上分支字符组成的字符串作为该叶子结点字符的编码

译码

译码的过程是分解电文中字符串，从根结点出发，按字符‘0’或‘1’确定找左孩子或右孩子，直到叶子结点，便求得该子串相应的字符