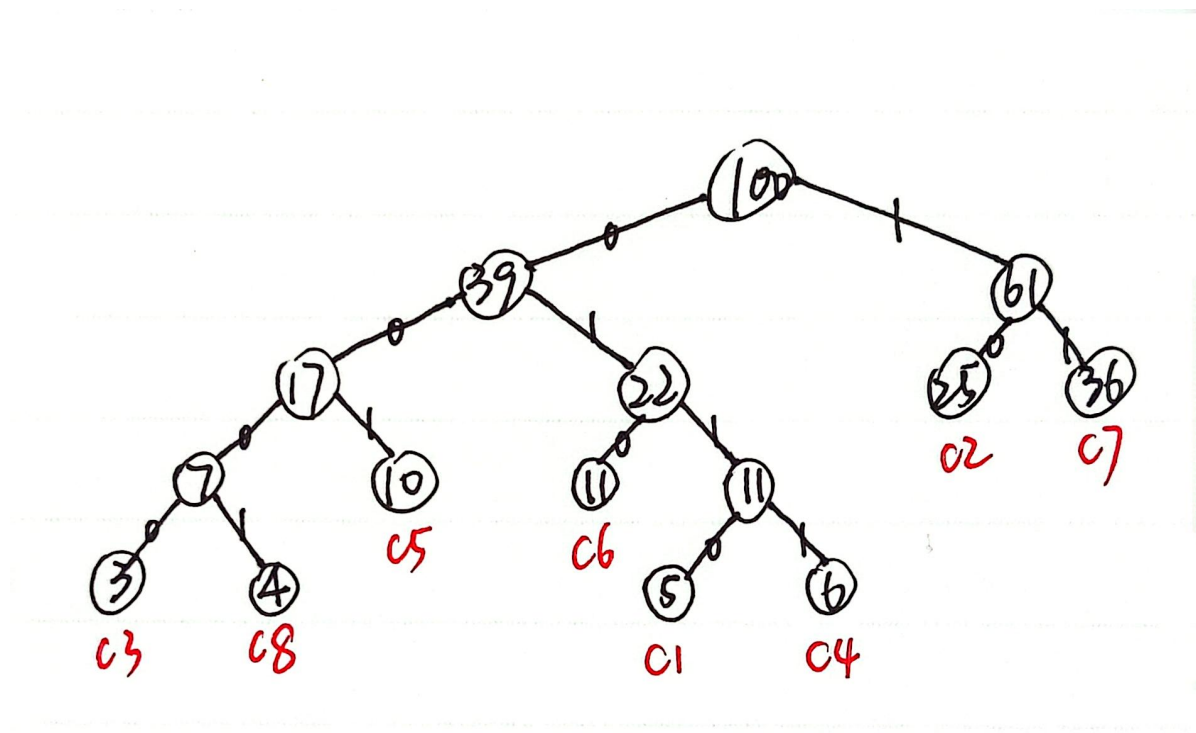


## 5.20

首先针对它们建立Huffman树



Huffman编码为

c1	c2	c3	c4	c5	c6	c7	c8
0110	10	0000	0111	001	010	11	0001

电文总码数为

$$4 \times 5 + 2 \times 25 + 4 \times 3 + 4 \times 6 + 3 \times 10 + 3 \times 11 + 2 \times 36 + 4 \times 4 = 257$$

## 5.23

(1)

```
template<class T>
int leaves(BinTreeNode<T>* t) const {
    if (t == NULL) return 0;
    if (t->leftChild == NULL && t->rightChild == NULL) return 1;
    return leaves(t->leftChild) + leaves(t->rightChild);
}
```

(2)

```

template<class T>
void exchange(BinTreeNode<T>* t) {
    if (t == NULL) return;
    BinTreeNode<T>* temp = t->leftChild;
    t->leftChild = t->rightChild;
    t->rightChild = temp;
    exchange(t->leftChild);
    exchange(t->rightChild);
}

```

## 5.29

```

template<class T>
void BinTree_Link_to_Array(BinTreeNode<T>* t, T a[], int n, int i)
{
    if (t == NULL) return;
    if (i < n)
    {
        a[i] = t->data;
        BinTree_Link_to_Array(t->leftChild, a, n, 2 * i + 1);
        BinTree_Link_to_Array(t->rightChild, a, n, 2 * i + 2);
    }
    else {
        cerr << "Subscript out of range" << endl;
        exit(1);
    }
}

```

## 5.33

### (1)

```

template<class T>
struct LsRCNode {
    T data;
    int llink, rlink;
};

template<class T>
struct Db1TagNode {
    T data;
    int ltag, rtag;
};

template<class T>
class LsRCTree
{
private:
    LsRCNode<T>* LsRCList;
    Db1TagNode<T>* Db1TagList;
    int MaxSize, CurrentSize;
public:
    LsRCTree(int size) :MaxSize(size), CurrentSize(0) {

```

```

        LsRcList = new LsRcNode<T>[size];
        Db1TagList = new Db1TagNode<T>[size];
        assert(LsRcList != NULL && Db1TagList != NULL);
    }
};

```

(2)

```

template<class T>
void LsRcTree<T>::Db1Tag_to_LsRc()
{
    stack<int> s;
    for (int i = 0; i < CurrentSize; i++) {
        if (Db1TagList[i].ltag == 0) {
            if (Db1TagList[i].rtag == 0) {
                LsRcList[i].llink = LsRcList[i].rlink = -1;
                int k = s.top();
                s.pop();
                LsRcList[k].rlink = i + 1;
            }
            else {
                LsRcList[i].llink = -1;
                LsRcList[i].rlink = i + 1;
            }
        }
        else {
            if (Db1TagList[i].rtag == 0) {
                LsRcList[i].llink == i + 1;
                LsRcList[i].rlink == -1;
            }
            else {
                LsRcList[i].llink = i + 1;
                s.push(i);
            }
        }
    }
}

```