20181026 数据结构作业

1800022769 张靖昆 20181026

我承诺诚实作业，没有抄袭他人！

作业 1：

1．画出其顺序存储和二叉链表存储

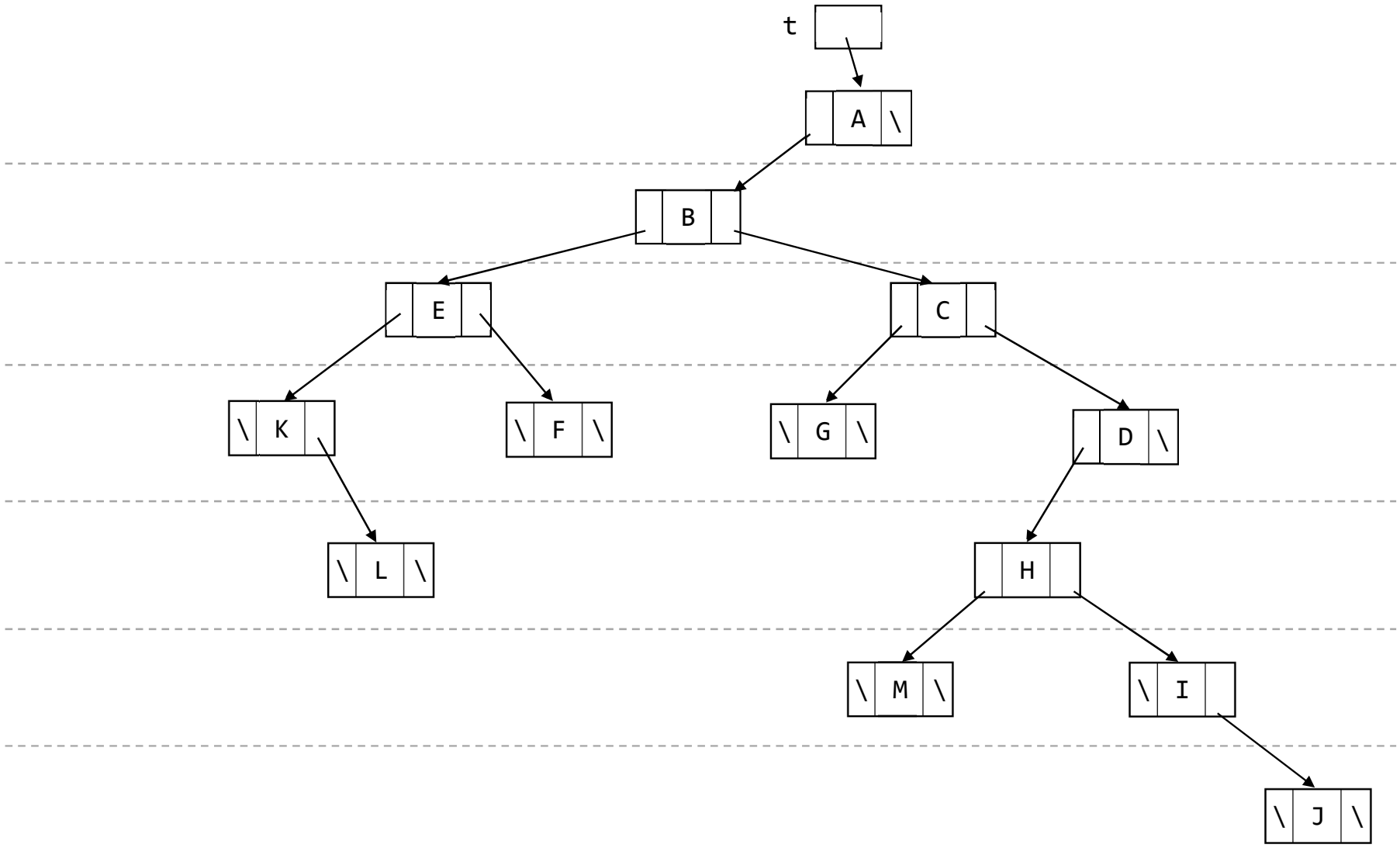
1）顺序存储

该二叉树的最大深度为 6，从而应当为其开辟 2^7-1=127 的数组长度才有可能满足其对连续存储空间的需求。**其中没有值的即代表对应结点值为空**。其中有值的结点，我使用**橙色底加粗**进行醒目标记。

下标	**0**	**1**	2	**3**	**4**	5	6	**7**	**8**	**9**	**10**	11	12	13	14	15	**16**	17	18	19
结点值	**A**	**B**		**E**	**C**			**K**	**F**	**G**	**D**						**L**			
下标	20	**21**	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
结点值		**H**																		

下标	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
结点值				M	I															
下标	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
结点值																				
下标	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
结点值											J									
下标	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
结点值																				
下标	120	121	122	123	124	125	126													
结点值																				

2）二叉链表存储


t
A
B
E
C
K
F
G
D
L
H
M
I
J

3）**列出该二叉树的叶子结点，并指出该二叉树的深度。**

叶子结点：L F G M J

深度：6

4）**分别写出该二叉树的先序、中序和后序遍历序列。**

前序：A B E K L F C G D H M I J

中序：K L E F B G C M H I J D A

后序：L K F E G M J I H D C B A

2．编写一个算法统计二叉树中叶子结点的个数

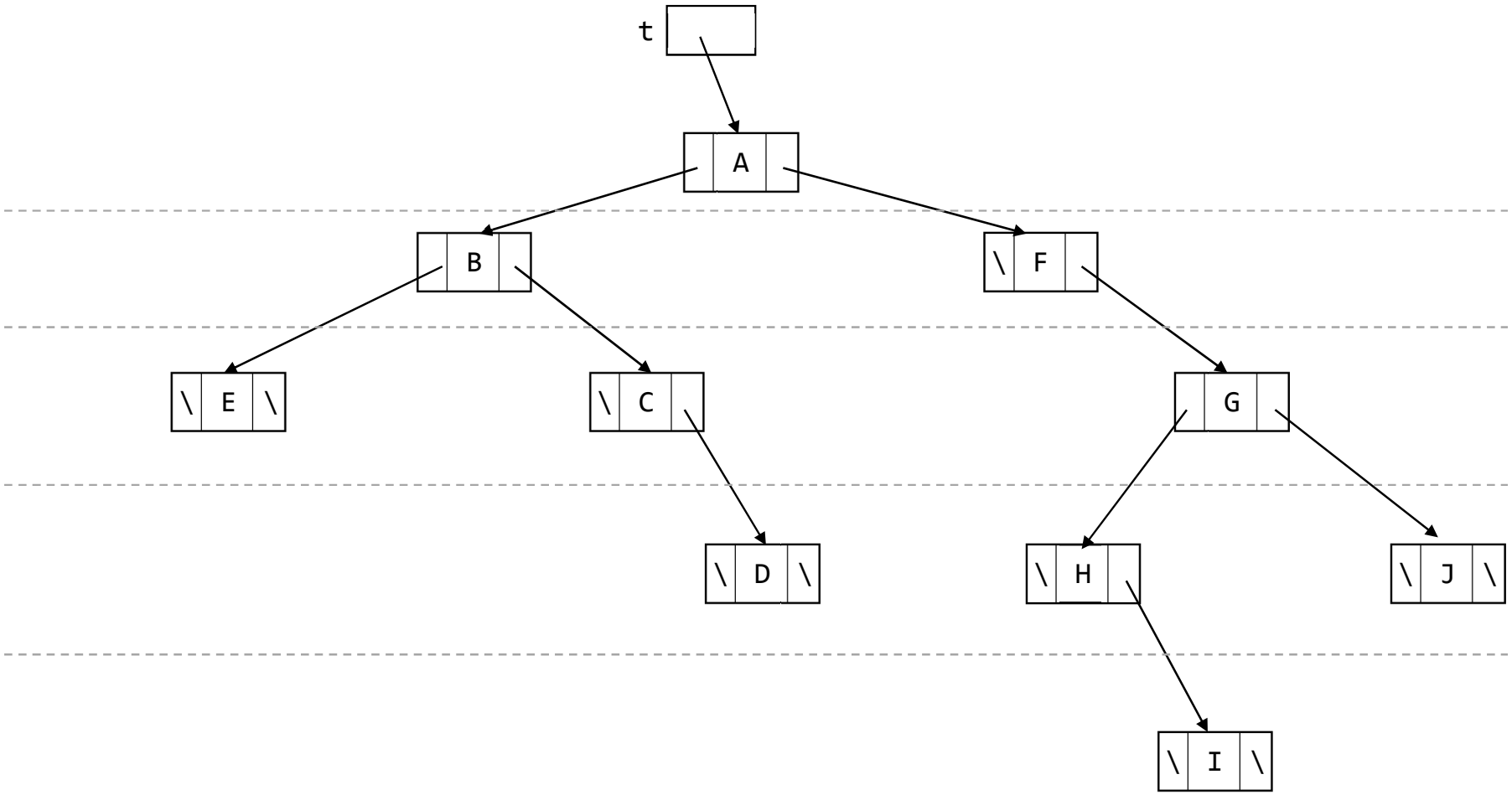
由于这道题比较简单，从而直接将 C 代码直接贴出来，本文采用前序遍历进行统计。这里不进行 main 函数与头文件的说明，仅对算法主体进行说明。

```
//定义二叉树结点
typedef struct BiTreeNode{
```

```
    char data; //结点数值
    struct BiTreeNode *lchild; //左孩子结点
    struct BiTreeNode *rchild; //右孩子结点
}BiTreeNode;
int count = 0;

void printCountOfLeaves(BiTreeNode *root){
    preOrder(root);
    printf("%d", count);
}

void preOrder(BiTreeNode *root){
    if(root == NULL) return;
    visit(root);
```

```
   preorder(root->lchild);
   preorder(root->rchild);
}

void visit(BiTreeNode *node){
   if(node->lchild == NULL && node->rchild == NULL)
      count++;
}
```

3．已知一颗二叉树的前序遍历结果是 ABECDFGHIJ，中序遍历结果是 EBCDAFHIGJ，试画出这颗二叉树。


t
A
B
\ F
\ E \
\ C
G
\ D \
\ H
\ J \
\ I \

作业 2:

1．试着找出分别满足系列条件的所有二叉树。

1）前序序列和中序序列相同

空二叉树，只有一个根结点的二叉树，非空非叶结点只有右孩子的二叉树

2）中序序列和后序序列相同

空二叉树，只有一个根结点的二叉树，非空非叶结点只有左孩子的二叉树

3）前序序列和后序序列相同

空二叉树，只有一个根结点的二叉树

4）前序、中序、后序序列都相同

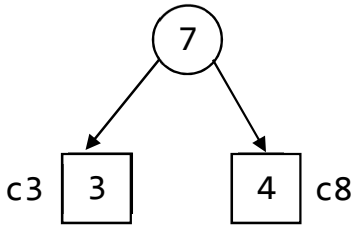
空二叉树，只有一个根结点的二叉树

2．假定用于通信的电文仅由 8 个字母 c1，c2，c3，c4，c5，c6，c7，c8 组成，各字母在电文中出现的频率分别为 5,25,3,6,10,11,36,4。试为这 8 个字母设计不等长 Huffman 编码，并给出该电文的总码数。

将频率按照升序进行排序得到如下顺序表

标号	1	2	3	4	5	6	7	8
频率	3	4	5	6	10	11	25	36

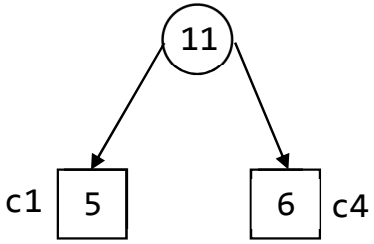
1）首先拿出频率最低的 2 个分别为 3 和 4 组成一颗扩充二叉树为



升序频率表更新为

标号	1	2	3	4	5	6	7
频率	5	6	7	10	11	25	36

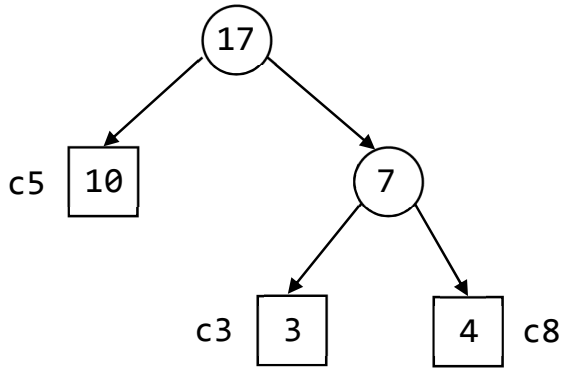
2）再拿出频率最低的 2 个分别为 5 和 6 组成一颗扩充二叉树为



升序频率表更新为

标号	1	2	3	4	5	6
频率	7	10	11	11	25	36

3）**再拿出频率最低的 2 个分别为 7 和 10 组成一颗扩充二叉树，其中频率为 7 的是第一步生成的扩充二叉树**

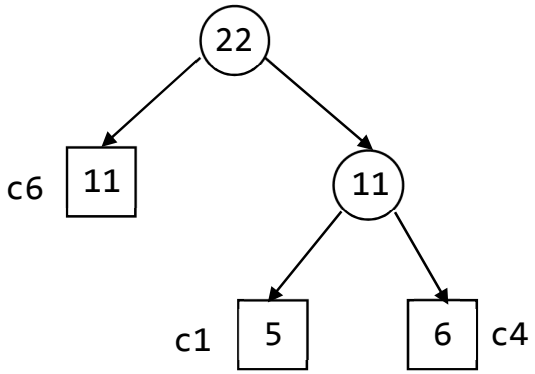


升序频率表更新为

标号	1	2	3	4	5

频率	11	11	17	25	36

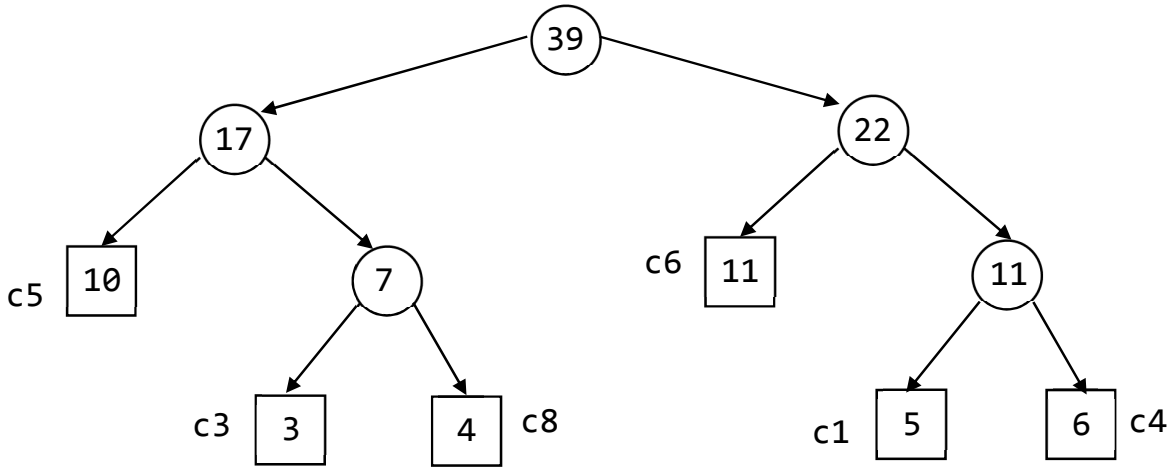
4）再拿出频率最低的 2 个分别为 11 和 11 组成一颗扩充二叉树，其中有一个 11 为第二步组成的扩充二叉树。



升序频率表更新为

标号	1	2	3	4
频率	17	22	25	36

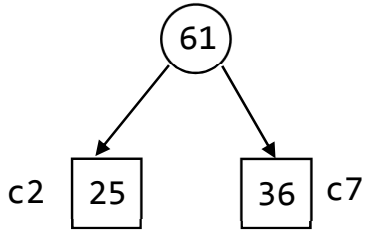
5）再拿出频率最低的 2 个分别为 17 和 22 组成一颗扩充二叉树，其中 17 为第三步形成的扩充二叉树，22 位第 4 步形成的扩充二叉树



升序频率表更新为

标号	1	2	3
频率	25	36	39

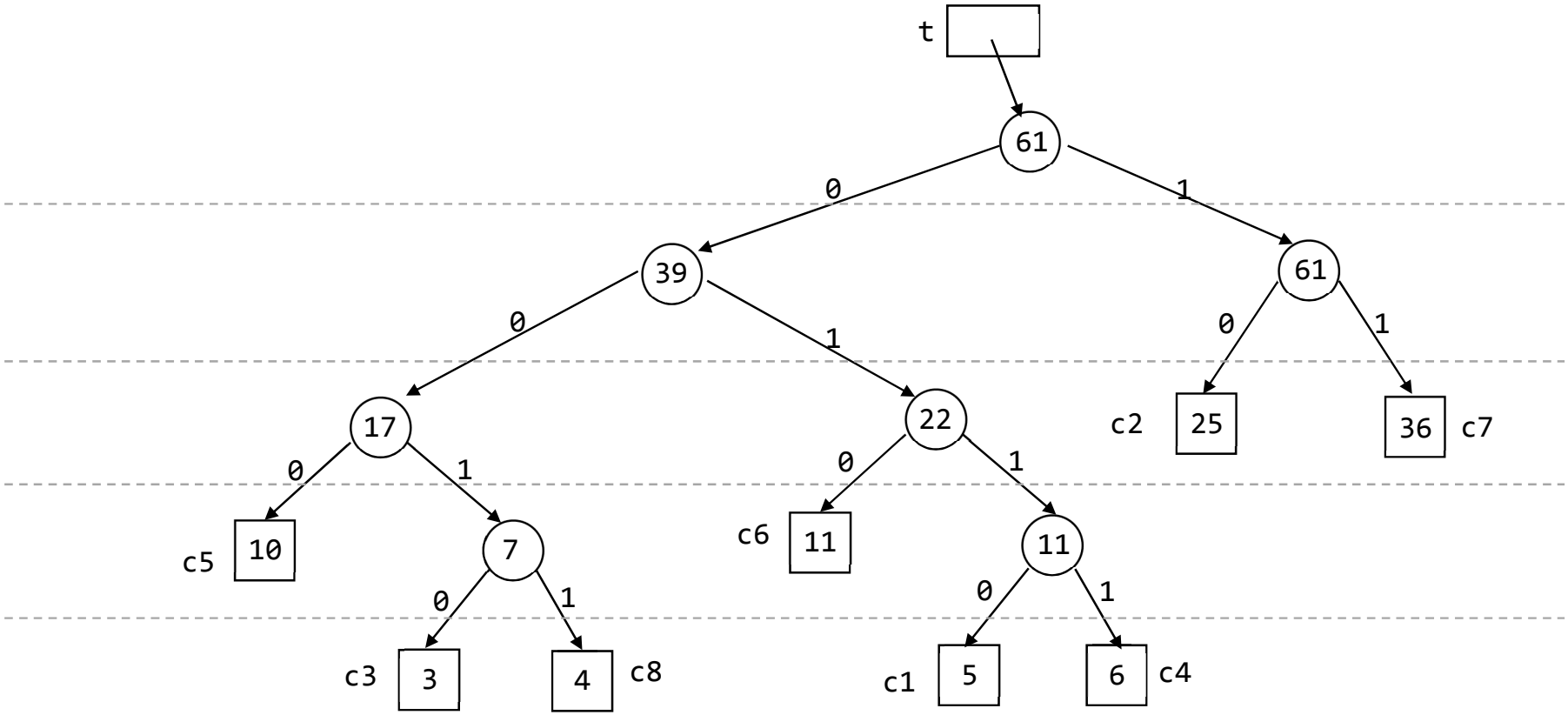
6）再拿出频率最低的 2 个分别为 25 和 36 组成一颗扩充二叉树。



升序频率表更新为

标号	1	2
频率	39	61

7）最后，将剩余的拼成最终的扩充二叉树即为赫夫曼树，它拥有最小的加权路径，令左子树路径编码为 0，右子树路径编码为 1，则得到如下赫夫曼编码树。



根据从根结点到各外结点的路径，从而得到如下对字符的编码表:

字符	编码
c1	0110
c2	10
c3	0010
c4	0111
c5	000
c6	010
c7	11
c8	0011

则该电文的总码数为 4×5 + 2×25 + 4×3 + 4×6 + 3×10 + 3×11 + 2×36 + 4×4 = 257.