本次作业题目为12，13，14，16，34，36

第12题：

typedef struct TreeNode

{

int left;

int right;

int level;

}TreeNode;

// Known from the question : TreeNode\* TreeArrays = (TreeNode\*)malloc(sizeof(TreeNode) \* n);

void Traverse(TreeNode\* TreeArrays){

assign(TreeArrays,TreeArrays[0],1);

}

void assign(TreeNode\* TreeArrays,TreeNode node,int level){

node.level = level;

if(node.left){

assign(TreeArrays,TreeArrays[node.left-1],level+1);

}

if(node.right){

assign(TreeArrays,TreeArrays[node.right-1],level+1);

}

}

第13题：

证明：

对于高度为h的满二叉树，所有的结点个数为2^h-1，其中根结点1个，叶结点2^(h-1)个，则内部结点为2^(h-1)-2个，满足题中结果。而其他高度为h的二叉树可以通过对该满二叉树依次删除底部结点得到。经过分析，可知，在满二叉树时，内部结点和叶结点树达到最多，则可证明之。

第14题：

typedef struct TreeNode

{

int left;

int right;

int lw;

int rw;

int mw;

}TreeNode;

// Known from the question : TreeNode\* TreeArrays = (TreeNode\*)malloc(sizeof(TreeNode) \* n);

int calMaxWeight(TreeNode\* TreeArrays,TreeNode node){

int max = 0;

if(node.left){

max = node.lw + calMaxWeight(TreeArrays,TreeArrays[node.left-1]);

}

if(node.right){

int t = node.rw + calMaxWeight(TreeArrays,TreeArrays[node.right-1]);

max = t > max ? t : max;

}

return max;

}

int addMaxWeight(TreeNode\* TreeArrays){

int i;

int count = sizeof(TreeArrays) / sizeof(TreeNode);

for(i=0;i<count;i++){

TreeArrays[i].mw = calMaxWeight(TreeArrays,TreeArrays[i]);

}

}

第16题：

typedef struct TreeNode

{

struct TreeNode\* left;

struct TreeNode\* right;

char element;

int mark;

}TreeNode;

// if mark == 0 ,this node is not be traversed. if 1,traversed.

void inOrderTraverse(TreeNode\* root){

while(root){

TreeNode\* p = root;

TreeNode\* q = root;

while(p->left!=NULL){

q = p;

p = p->left;

} // 令p为最左儿子，q为p的父亲 或者p,q均为root。

if(!p->mark){

printf("%c",element);

mark = 1;

} // 输出p的值，并标记为已读

if(p->right){

p->left = p->right;

p->right = NULL; // 如果p有右子树，则令有右子树转换为左子树

}else{

q->left = NULL;

free(p); // 如果没有左右子树，则删除该结点

}

}

}

第34题：

#define MAX 1000

typedef struct TreeNode

{

char element;

struct TreeNode\* left;

struct TreeNode\* right;

}TreeNode;

char\* Traverse(TreeNode\* root){

char pc[MAX];

if(!root){

pc[0] = 0;

}else if(!(root->left)&&!(root->right)){

sprintf(pc,"(%c)",root->element);

}else{

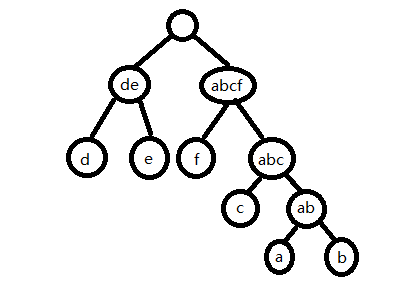
sprintf(pc,"(%c:%s,%s)",root->element,Traverse(root->left),Traverse(root->right));

}

return pc;

}

第36题：



a:1110

b:1111

c:110

d:00

e:01

f:10

平均编码长度：(0.07+0.09)\*4+0.12\*3+(0.22+0.23+0.27)\*2=2.44