

习题 3 树结构

申昱 22920192204266

3-1 C

3-2 D

3-3

$$n_0 = 1 + 0 \cdot n_1 + 1 \cdot n_2 + 2 \cdot n_3 + \dots + (K-1) \cdot n_K$$

3-4

$$n = n_0 + n_1 + n_2$$

$$n = n_1 + 2n_2$$

得： $n_0 = n_2 + 1$

3-5 B

3-6 C

3-7 B

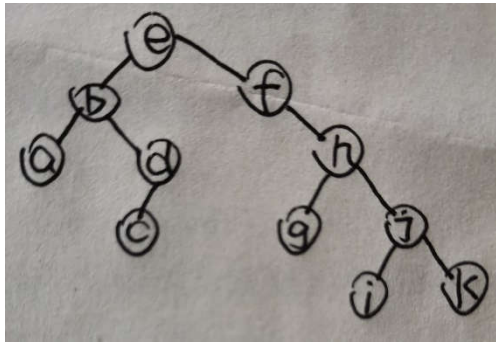
3-8 A

3-9

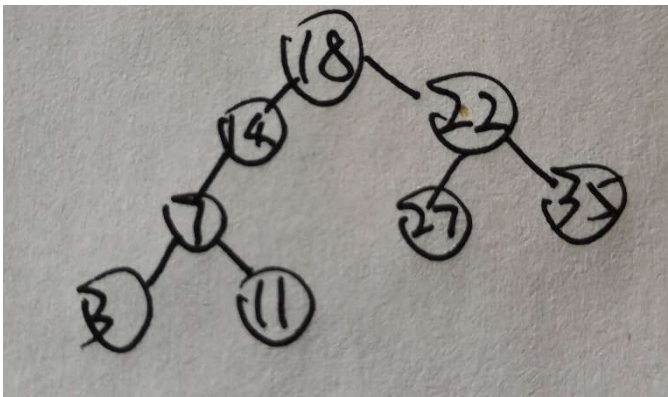
如果前序序列和中序序列都为空，那么构造一棵空树。否则

- 1、根据前序可确定根。
- 2、根据根和中序，可以确定左子树集合和右子树集合，并得到左子树中序序列和右子树中序序列。
- 3、在前序序列中划分出左子树前序序列和右子树前序序列。
- 4、根据左子树前序序列和左子树中序序列构造左子树。
- 5、根据右子树前序序列和右子树中序序列构造右子树。

3-10



3-11



3-12

```
int leaf(BiTree T)
{
    if(!T)
        return 0;
    if(!T->Lchild && !T->Rchild)
        return 1;
    return leaf(T->Lchild)+leaf(T->Rchild);
}
```

3-13

```

int exchange(BiTree T)
{
    BiTree temp =new BiTnode;
    if(!T)
        return;
    if(!T->Lchild && !T->Rchild)
        return;
    temp=T->Lchild;
    T->Lchild=T->Rchild;
    T->Rchild=temp;
    exchange(T->Lchild);
    exchange(T->Rchild);

}

```

3-14 B

3-15

```

TNode* SearchNodeBehind(BiTree *T, TNode *p)
{
    return p->Rchild;
}

```

3-16 A

3-17

```
void GetParent (BiTree *T, TNode *child, TNode * &parent)
{
    if (!T)
        return ;
    if (T->leftChild==child || T->rightChild==child)
    {
        parent=T;
        return ;
    }
    GetParent (T->leftChild, child, parent);
    GetParent (T->rightChild, child, parent);
    return ;
}
```