

5.2

- (1) 无序树：9 棵
- (2) 有序树：12 棵
- (3) 二叉树：30 棵

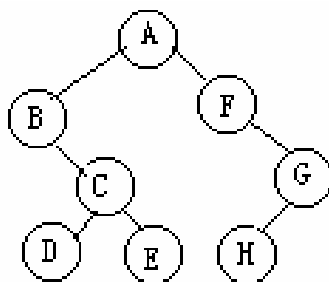
5.4

- (1)  $2^{k-1}$
- (2)  $\left\lceil \frac{i-1}{k} \right\rceil$
- (3)  $k(i-1) + m + 1$
- (4)  $i + 1 \leq \left\lceil \frac{i-1}{k} \right\rceil k + 1$

5.5

- (1) 空二叉树和所有结点均无左孩子的二叉树
- (2) 空二叉树和只有一个根
- (3) 空二叉树和所有结点均无右孩子

5.6



5.7

先：DEHFJGCKAB  
中：HEJFGKCDAB  
后：HJKCGFEBAD

5.9

(1)

```
template <class T>
void BinaryTree<T>::Del(BTNode<T> *p)    //private
{
    if (p!=NULL)
    {
        Del(p->lChild);
        Del(p->rChild);
        delete p;
    }
}
template <class T>
```

```

void BTree<T>::Del() //public
{
    Del(root);
    root = NULL;
}

```

(2)

/\*求二叉树中度为 1 的结点个数\*/

```

template <class T>
int BinaryTree<T>::CountDegree1()
{
    int total = 0;
    CountDegree1(root, total);
    return total;
}

template <class T>
void BinaryTree<T>::CountDegree1(BTNode<T> *t, int &num)
{
    if (t)
    {
        if ( ( (t->lChild != NULL) && (t->rChild == NULL) ) ||
            ( (t->lChild == NULL) && (t->rChild != NULL) ) )
        {
            ++num;
        }
        CountDegree1(t->lChild, num);
        CountDegree1(t->rChild, num);
    }
}

```

(3)

```

template <class T>
void BinaryTree<T>::Exch(BTNode<T> *p) //private
{
    if (p!=NULL)
    {
        BTNode<T> *temp;
        temp=p->lchild;
        p->lchild=p->rchild;
        p->rchild=temp;
        Exch(p->lchild);
        Exch(p->rchild);
    }
}

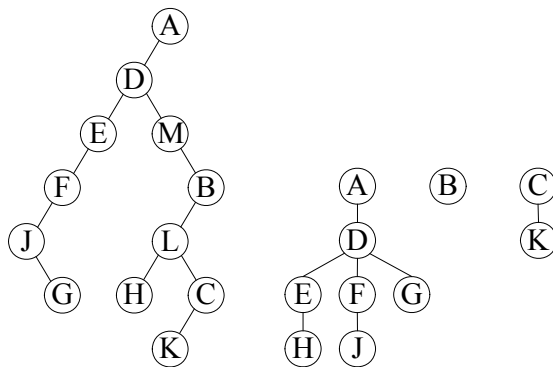
```

```

    }
}
template <class T>
void BTree<T>::Exchange() //public
{
    Exch(root);
}

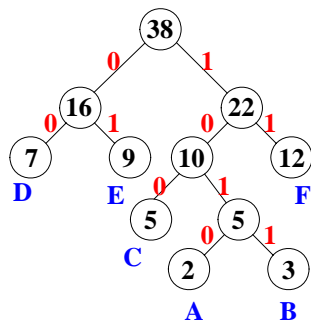
```

5.14



5.19

(1)



(2) WPL = 91

(3) 各字符的编码

A:1010

B:1011

C:100

D:00

E:01

F:11