

5.23 (1) 统计二叉树中叶结点的个数

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

(2) 交换每个结点的左右子女

```
template<class T>
void BinTreeNode<T>::Exchange(BinTreeNode<T> *subTree) {
    BinTreeNode<T> *p;
    if (subTree->leftChild != NULL || subTree->rightChild != NULL) {
        p = subTree->leftChild;
        subTree->leftChild = subTree->rightChild;
        subTree->rightChild = p;
        Exchange(subTree->leftChild);
        Exchange(subTree->rightChild);
    }
}
```

5.26 查找二叉树中值为x的结点并打印其所有祖先结点

```
template<class T>
int BinTreeNode<T>::PrintParent(BinTreeNode<T> *subTree, const T& x) {
    if (subTree == NULL) return 0;
    if (subTree->data == x) return 1;
    if (PrintParent(subTree->leftChild, x) || PrintParent(subTree->rightChild, x)) {
        cout << subTree->data << endl;
        return 1;
    }
    return 0;
}
```