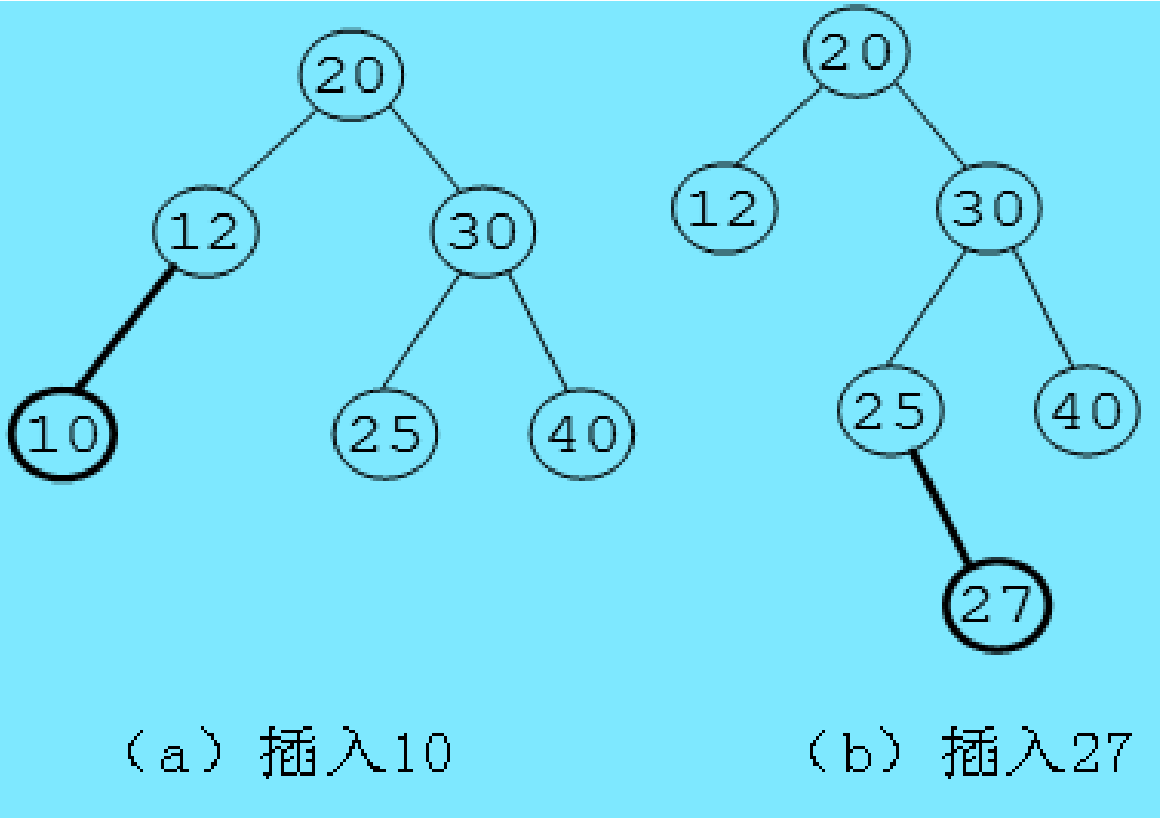
5.4 二叉排序树的插入 BST--insert

```
if ( 二叉排序树中查不到该结点)
    {   建立新结点;
        if (原二叉排序树是空树)
            新结点作为根结点;
         else if (新结点<父结点)
                    插入左子树;
               else  插入右子树;
    }
```



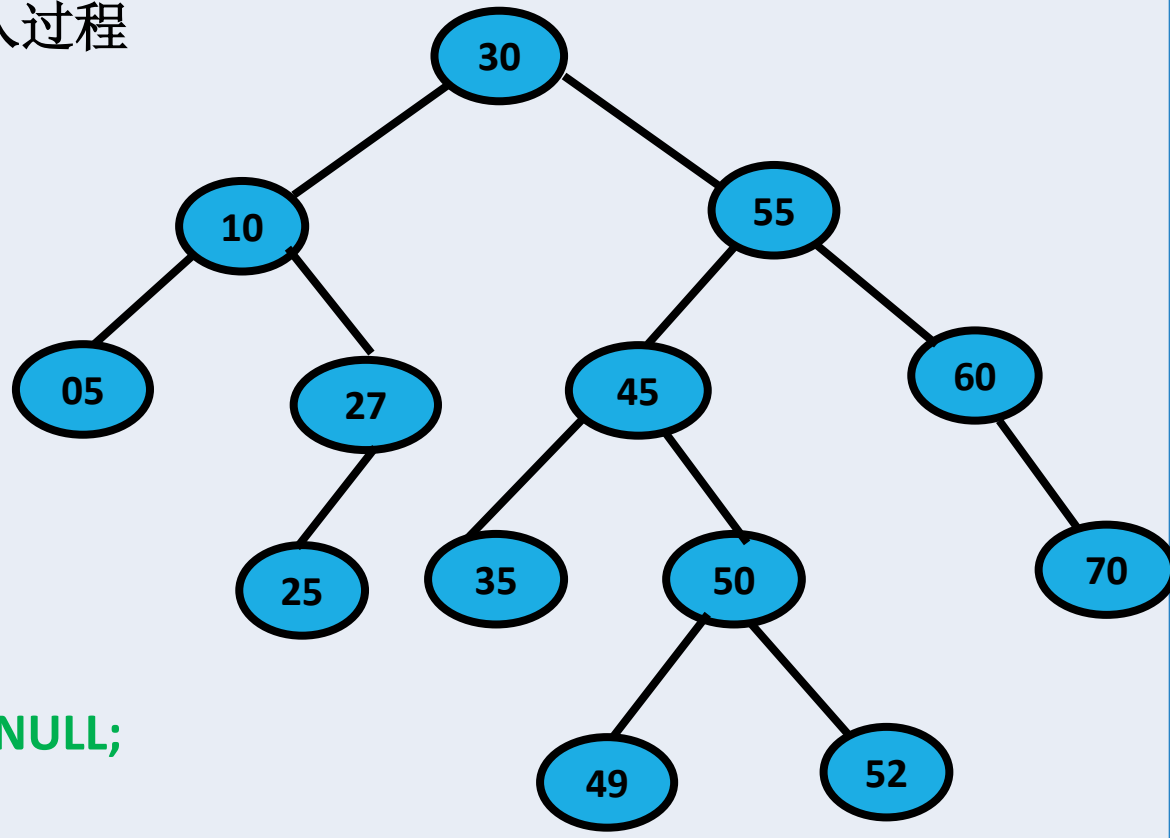
5.4 二叉排序树的插入 BST--insert 算法5-2

```
1       int BSTInsert(BinSearTree bt, DataType key) //二叉排序树插入过程
2       {
3          BSTreeNode p, temp;
4          temp = BSTSearch(bt, key); //调用查找算法5-1
5          if ( temp == NULL) {
6/7                              printf("exist this key\n"); return 0;
8                  }
9          p = (BSTreeNode*)malloc(sizeof(struct BinSearTreeNode));//申请结点
10         if (p == NULL) {
11/12              printf("Alloc Failure!\n"); return 0;
13         }
14/15      p->data = key; //数据域赋值    p->leftchild = p->rightchild = NULL; //指针域赋值
16         if (key < temp->data)
17                 temp->leftchild = p;  //作为左孩子插入
18          else
19                 temp->rightchild = p; //作为右孩子插入
20        return 1;
21      }
```

5.4 二叉排序树的插入 BST--insert

算法5-2

```
int BSTInsert(BinSearTree bt, DataType key) //二叉排序树插入过程
{
   BSTreeNode p, temp;
   temp = BSTSearch(bt, key); //调用查找算法5-1
   if ( temp == NULL) {
                        printf("exist this key\n"); return 0;
            }
   p = (BSTreeNode*)malloc(sizeof(struct BinSearTreeNode));
   if (p == NULL) {
            printf("Alloc Failure!\n"); return 0;
   }
   p->data = key; //数据域赋值    p->leftchild = p->rightchild = NULL;
   if (key < temp->data)
            temp->leftchild = p;  //作为左孩子插入
    else
            temp->rightchild = p; //作为右孩子插入
  return 1;
}
```



二叉排序树 的生成过程示例

若从空树出发，经过一系列插入操作后，可生成一棵二叉排序树。
例如，K={18, 73, 10, 05, 68, 99, 27, 41, 51, 32, 25}