



本节主题:

二叉排序树

二叉排序树

名称

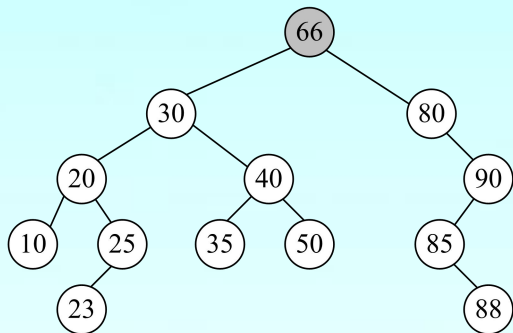
二叉排序树(简称BST)又称二叉查找(搜索)树

定义

二叉排序树或者是空树，或者是满足如下性质

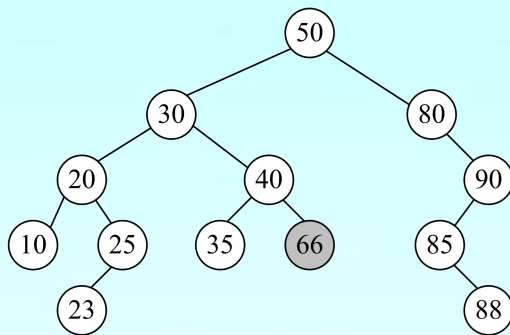
- (1) 若它的左子树非空，则左子树上所有记录的值均小于根记录的值；
- (2) 若它的右子树非空，则右子树上所有记录的值均大于根记录的值；
- (3) 左、右子树本身又各是一棵二叉排序树。

二叉排序树



//二叉排序树的存储结构

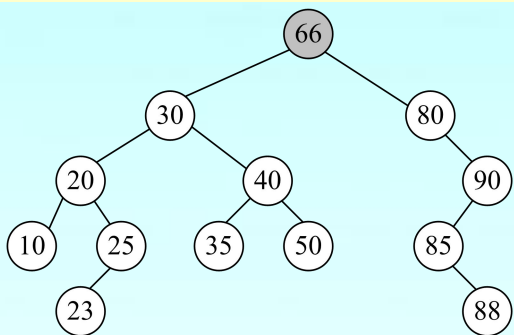
```
typedef struct node
{
    KeyType key;
    InfoType data;
    struct node *lchild,*rchild;
} BSTNode;
```



非二叉排序树

二叉排序树上的查找

```
BSTNode *SearchBST(BSTNode *bt,KeyType k)
{
    if (bt==NULL || bt->key==k)
        return bt;
    if (k<bt->key)
        return SearchBST(bt->lchild,k);
    else
        return SearchBST(bt->rchild,k);
}
```

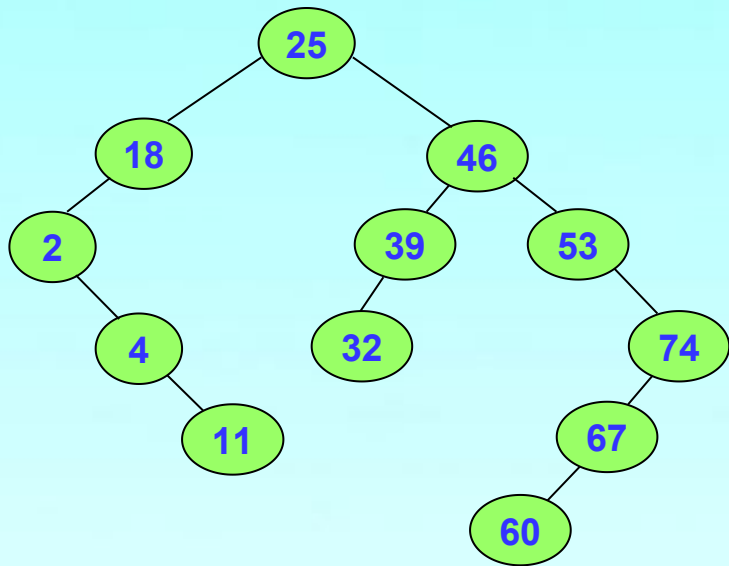


对二叉排序树中序遍历的惊人发现？！

```
BSTNode *SearchBST1(BSTNode *bt,KeyType k)
{
    while (bt!=NULL)
    {
        if (k==bt->key)
            return bt;
        else if (k<bt->key)
            bt=bt->lchild;
        else
            bt=bt->rchild;
    }
    return NULL;
}
```

二叉排序树的查找一定是 $O(\log_2 n)$ ？

二叉树的生成



```
BSTNode *CreateBST(KeyType A[],int n)
{
    BSTNode *bt=NULL;
    int i=0;
    while (i<n)
    {
        InsertBST(bt,A[i]);
        i++;
    }
    return bt;
}
```

```
int main()
{
    BSTNode *bt;
    int n=12;
    KeyType a[]={25,18,46,2,53,39,32,4,74,67,60,11};
    bt=CreateBST(a,n);
    .....
}
```

二叉排序树中节点的插入

问题

在二叉排序树T中插入一个关键字为k的新记录，要保证插入后仍满足BST性质

插入过程：

(1) 若二叉排序树T为空，则创建一个key域为k的节点，将它作为根节点；

(2) 否则将k和根节点的关键字比较

- 若两者相等，则说明树中已有此关键字k，无须插入，直接返回0；
- 若 $k < T \rightarrow key$ ，则将k插入根节点的左子树中；
- 否则将它插入右子树中。

//在以*p为根节点的BST中插入一个关键字为k的节点。成功返回1，否则返回0。

```
int InsertBST(BSTNode *&p,KeyType k)
{
    if (p==NULL)
    {
        p=(BSTNode *)malloc(sizeof(BSTNode));
        p->key=k;
        p->lchild=p->rchild=NULL;
        return 1;
    }
    else if (k==p->key)
        return 0;
    else if (k<p->key)
        return InsertBST(p->lchild,k);
    else
        return InsertBST(p->rchild,k);
}
```

任何节点插入到二叉排序树时，都是以叶子节点插入的。

例 排序二叉树的生成及查找性能分析

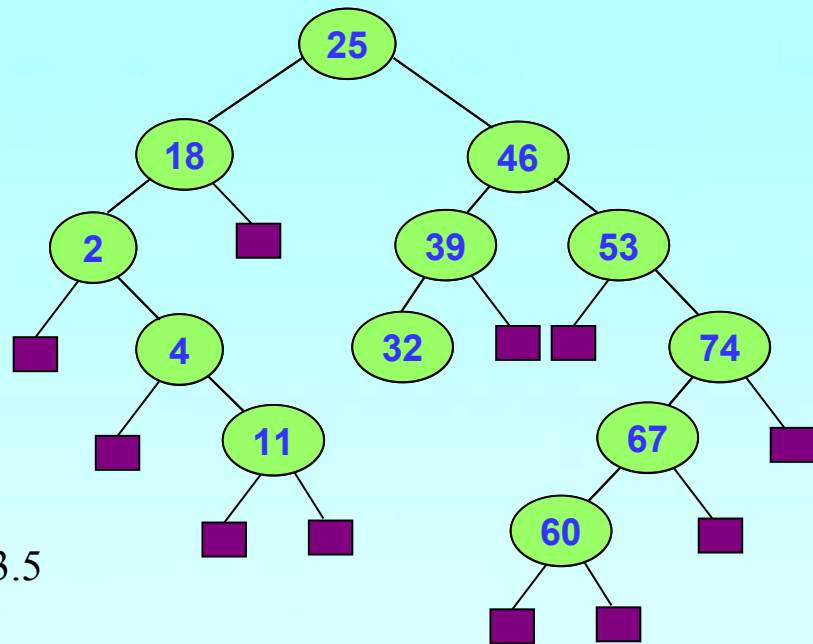
问题

- 已知一组关键字为{25,18,46,2,53,39,32,4,74,67,60,11}
- 按表中的元素顺序依次插入到一棵初始为空的二叉排序树中，画出该二叉排序树
- 求在等概率的情况下查找成功的平均查找长度。

解

$$ASL_{成功} = \frac{1 \times 1 + 2 \times 2 + 3 \times 3 + 3 \times 4 + 2 \times 5 + 1 \times 6}{12} = 3.5$$

$$ASL_{不成功} = \frac{1 \times 2 + 3 \times 3 + 3 \times 4 + 3 \times 5 + 2 \times 6}{11} = 4.18$$



例 排序二叉树的生成及查找性能分析(续)

