No.
Date

二叉排序树

1. 目的：节省排序时间，节省检索时间。
（不是排成有序表）　（二分搜索思想）

2. 性质：若非空，则左子树所有结点关键码小于根结点关键码，右子树所有结点关键码大于根结点关键码。

3. 算法：

① 检索

```
int search (PBinSearchTree ptree, keyType key, PBinSearchNode *position)
{ p=*ptree; q=p;                          (q记录p的父结点)
  while (p!=NULL){
    q=p;
    if (p->key==key){*position=p; return 1;}
    else if (p->key>key) p=p->llink;
    else p=p->rlink;
  }
  *position=q; return 0;
}                                          (与二分法检索相似).
```

② 插入结点

```
int insert (PBinSearchTree ptree, keyType key)
{ ① if (Search(ptree, key, &position)==1) return 1;   (已经有了)
  ② p=malloc(sizeof(BinSearchNode));
    p->key=key; p->llink=p->rlink=NULL;               (本身结点)
  ③ if (position==NULL) *ptree=p;                     (原树为空).
    else if (position->key>key) position->llink=p;
    else position->rlink=p;
    return 1;
}
```

③ 构造二叉排序树　　返值参数

```
int createSearchTree(PBinSearchTree ptree, SeqDictionary *dic)
{   *ptree = NULL;
    for (i=0; i<dic->n; i++)
        if(!insert(ptree, dic->element[i])) return 0;   (没有足够空间建树时)
    return 1;
}
```

④ 删除元素

描述：(1) 找到结点 p，待删除。(同时找到其父结点)

(2) 若p无左子树，则用p的右子女代替p.

(3) 否则，在p的左子树中找到关键码最大的r（一直向右走），用r结点代替 p，并用r原来的左子女代替r。

~~代码：int delete(PBinSearchTree ptree, keyType key).~~

4. 图示　　k={18, 10, 5, 99, 41, 32}.