



上海工程技术大学

实验报告

实验四 查找

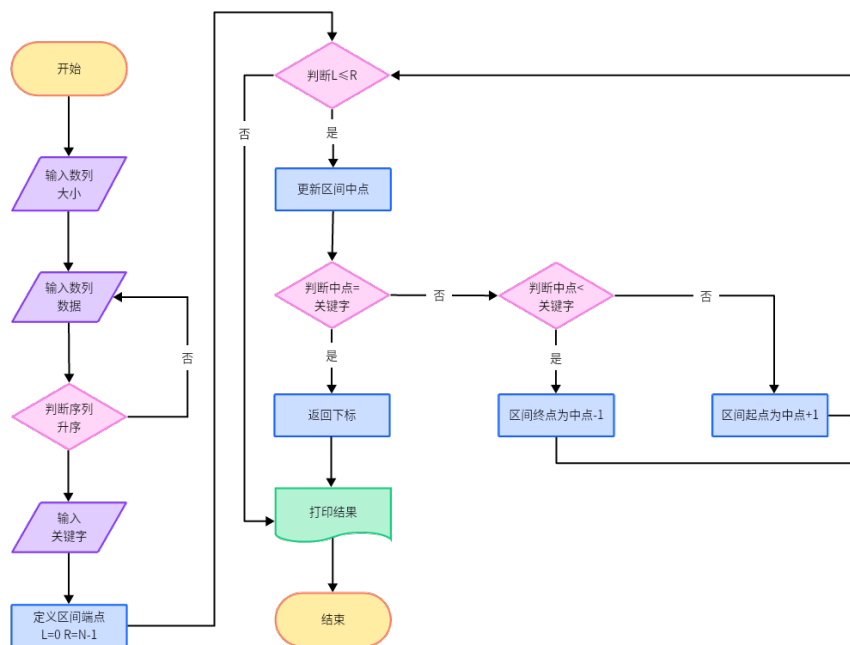
1. 题目 1

在关键字有序序列中采用折半查找法查找关键字为给定值 k 的元素，输出查找结果。

要求：

有序序列和给定值 k 都从键盘输入。

1.1. 流程图



1.2. 代码

```
#include <stdio>
#include <stdlib>
typedef int ElementType;
typedef struct LNode *List;
```

```

struct LNode{
    ElementType Data[20];
    int last;
};

int BinSearch(List L, ElementType X) {
    int left;
    int right;
    int mid;
    if(L->last) {
        left=0;
        right=L->last;
        while(left<=right) {
            mid=(left+right)/2;
            if(X==L->Data[mid]) {
                return mid;
            }else if(X>L->Data[mid]) {
                left=mid+1;
            }else if(X<L->Data[mid]) {
                right=mid-1;
            }
        }
        return -1;
    }
    return -1;
}

List MakeEmpty() {
    List PtrL;
    PtrL=(List)malloc(sizeof(struct LNode));
    PtrL->last=-1;
    return PtrL;
}

void Insert(List PtrL, int length) {
    int i;
    ElementType X;
    printf("读入序列的元素\n");
    for(i=0;i<length;i++) {
        scanf("%d",&X);
        PtrL->Data[i]=X;
        PtrL->last++;
    }
}

void OrderCheck(List &L, int length)
{

```

```

    int i=1;
    while(i<L->last)
    {
        if(L->Data[i-1]>=L->Data[i])
        {
            printf("非升序\n");
            printf("重新");
            Insert(L, length);
            break;
        }
        i++;
    }
}
int main() {
    List PtrL;
    PtrL=MakeEmpty();
    int length;
    printf("读入序列的长度\n");
    scanf("%d",&length);
    Insert(PtrL, length);
    OrderCheck(PtrL, length);
    ElementType target;
    printf("读入需要寻找的元素\n");
    scanf("%d",&target);
    int position;
    position=BinSearch(PtrL, target);
    if(position!=-1) {
        printf("该元素在第%d 位\n", position+1);
    }else{
        printf("没有找到\n");
    }
    return 0;
}

```

1.3. 结果

读入序列的长度

5

读入序列的元素

3 1 2 7 8

非升序

重新读入序列的元素

1 2 3 7 8

读入需要寻找的元素

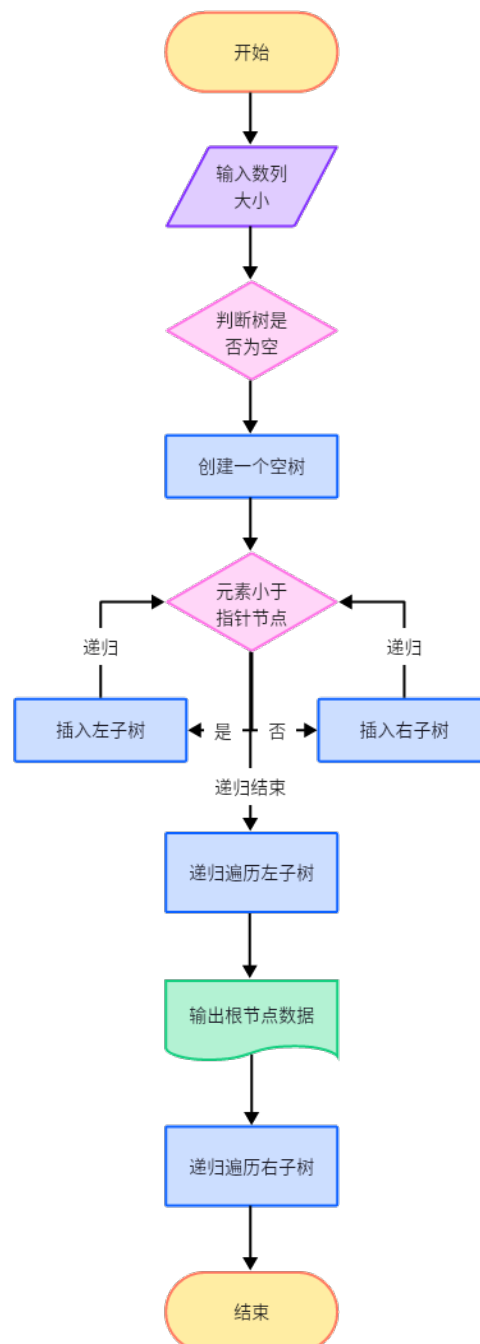
8

该元素在第5位

2. 题目 2

给定关键字序列为{16, 5, 17, 29, 11, 3, 15, 20}，按表中元素的顺序依次插入，建立相应的二叉排序树，给出其中序序列。

2.1. 流程图



2.2. 代码

```
#include <stdio>
#include <stdlib>
```

```

typedef int ElementType;
typedef struct BinSearchTreeNode{
    ElementType data;
    struct BinSearchTreeNode *left;
    struct BinSearchTreeNode *right;
}BSTNode,*PNode;
PNode insertBST(ElementType data,PNode root){
    if(root == NULL)
    {
        root = (PNode)malloc(sizeof(BSTNode));
        root->data = data;
        root->left = NULL;
        root->right = NULL;
        printf("%d ",data);
        return root;
    }
    if(data < root->data)
        root->left = insertBST(data,root->left);
    else if(data > root->data)
        root->right = insertBST(data,root->right);
    return root;
}
void print(PNode root){
    if(root!=NULL){
        print(root->left);
        printf("%d ",root->data);
        print(root->right);
    }
}
int main(){
    ElementType data[8]={16,5,17,29,11,3,15,20};
    int i;
    PNode root=NULL;
    for(i=0;i<8;i++){
        root=insertBST(data[i],root);
    }
    printf("全部入树\n");
    printf("中序序列为:\n");
    print(root);
    printf("\n");
    return 0;
}

```

2.3. 结果

16 5 17 29 11 3 15 20 全部入树

中序序列为：

3 5 11 15 16 17 20 29

3. 实验小结

我们学习了折半查找法，在一个有序序列中快速地查找到给定值的元素。它利用了有序序列的特性，通过每次将查找范围缩小一半，从而提高查找效率。在实际应用中，折半查找法是一种非常常见的算法。

我们还学习了二叉排序树，在建立二叉排序树时，我们可以按照顺序依次插入元素，这样就可以保证树的有序性质。在实际应用中，二叉排序树可以用于快速地进行查找、插入和删除操作。