

# 实验六

# 折半查找实验



# 一、实验目的

- 掌握折半查找算法的基本思想
- 掌握折半查找算法的实现方法
- 掌握折半查找的时间性能
- 掌握折半查找类的定义和使用



## 二、实验要求

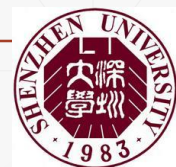
- 熟悉C++语言编程
- 了解折半查找的原理
- 了解折半查找类的定义、应用



# 三、实验内容

## 1、问题描述

- 在一个有序序列中，折半查找一个关键字
- 返回查找是否成功，如果成功，输出关键字所在的位置和查找次数



# 三、实验内容

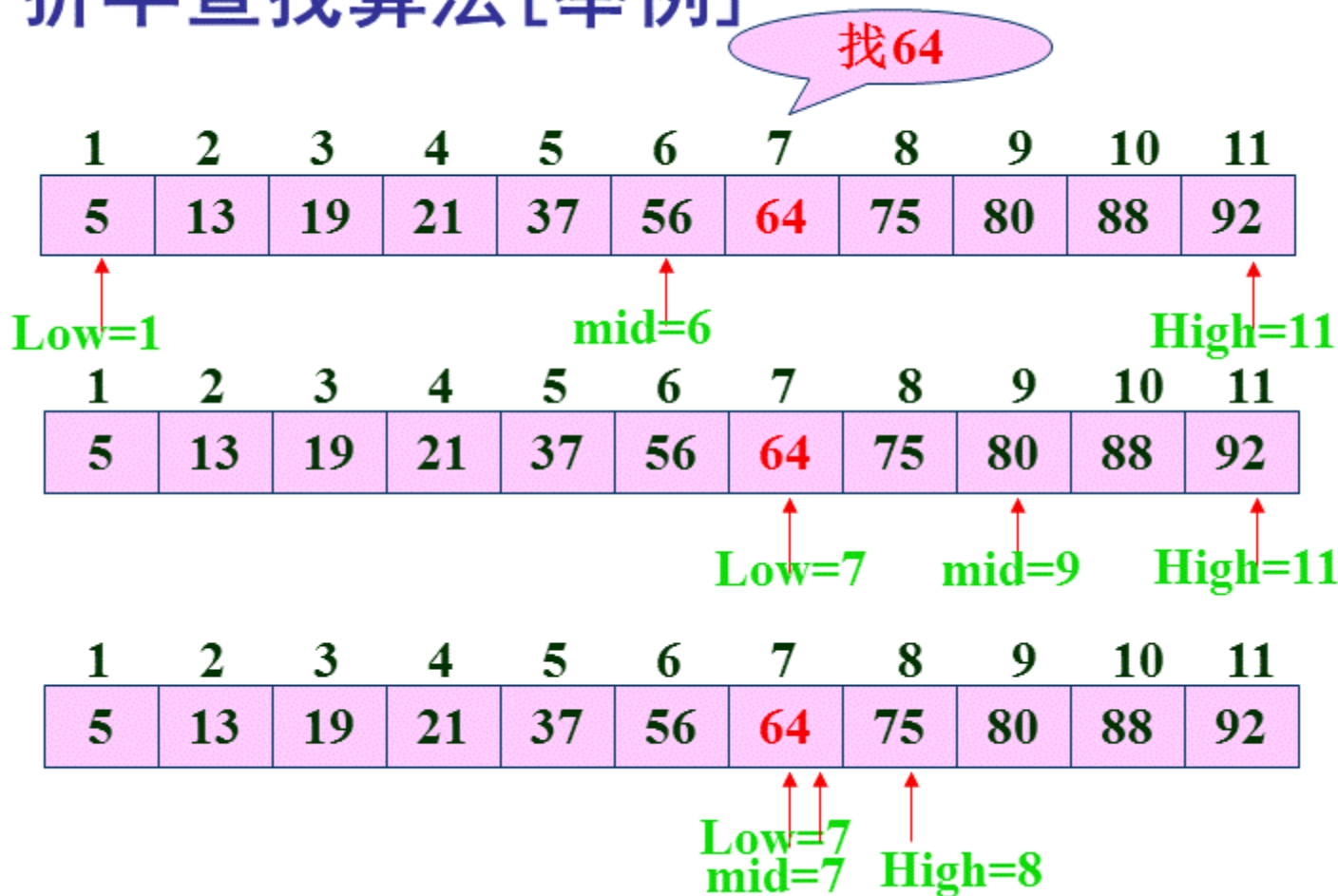
## 2、折半查找算法

1.  $n$ 个对象从小到大存放在有序顺序表ST中， $k$ 为给定值
2. 设low、high指向待查元素所在区间的下界、上界，  
即low=1, high= $n$
3. 设mid指向待查区间的中点，即 $mid = (low + high) / 2$
4. 让 $k$ 与mid指向的记录比较  
若 $k = ST[mid].key$ ，查找成功，结束  
若 $k < ST[mid].key$ ，则 $high = mid - 1$  [上半区间]  
若 $k > ST[mid].key$ ，则 $low = mid + 1$  [下半区间]
5. 重复3, 4操作，直至 $low > high$ 时，查找失败。



# 三、实验内容

## 2、折半查找算法[举例]



# 三、实验内容

## 3、输入

- 第一行：测试次数。
- 每个样本分2行：
  - 第一行：第一个数字 $n$ 表示样本数目，其后跟 $n$ 个样本；
  - 第二行：查找的关键字的值。



# 三、实验内容

## 4、输入样本

2

5 2 3 4 5 7

4

6 1 2 3 4 6 8

7



# 三、实验内容

## 5、输出

- 查找是否成功(1—表示成功, 0表示不成功)
- 所在位置(0—表示不成功)
- 查找次数



# 三、实验内容

## 6、输出样本

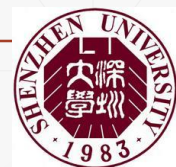
1 3 1

0 0 3



# 四、实验步骤

- 1、折半查找变量的定义
- 2、生成顺序有序表函数
- 3、折半查找函数
- 4、主程序



# 四、实验步骤

## 1、折半查找变量的定义

定义折半查找所有用到的变量：

int BinSuccess;           //查找是否成功（1—成功，0—不成功）

int BinPos;                //查找位置（0表示不成功）

int BinCount;             //查找次数

int BinList[32];          //有序表

int BinListLen;          //有序表长度

};



# 四、实验步骤

## 2、生成顺序表

利用直接插入排序，将输入的关键字排成有序序列

```
void CreateSequence(int *r, int n)
{
    int i, j, temp;
    BinListLen = n;
    for (i=1; i<n; i++) { //利用直接插入排序将顺序表元素排成升序序列
        if (r[i+1] < r[i]) {
            temp = r[i+1];
            for (j=i; j>=1; j-=1)
                if (temp<r[j]) r[j+1] = r[j]; else break;
            r[j+1] = temp;
        }
        for (i=1; i<=n; i++) BinList[i] = r[i]; // 数据放到有序顺序表中
    }
```



# 四、实验步骤

## 3、折半查找

```
int BinSearchKey(int Key)
```

```
{ int Low, Mid, High;
```

```
    Low = 1;
```

```
    //low指向待查元素所在区间的下界
```

```
    High = BinListLen;
```

```
    //high指向待查元素所在区间的上界
```

```
    BinSuccess = 0;    BinPos = 0;    BinCount = 0;
```

```
    :
```

```
    return(BinCount);
```

```
    // 返回查找次数
```

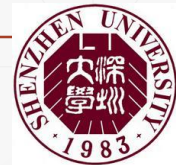
```
}
```



# 四、实验步骤

## 4、主程序

```
int main()
{ int r[32], i, j, Key, TestNum, SampleNum;
  cin >> TestNum; // 输入测试次数
  for (i=0; i<TestNum; i++) {
    cin >> SampleNum; // 输入样本数目
    for (j=1; j<=SampleNum; j++) cin >> r[j]; // 输入样本数据
    CreateSequence(r, SampleNum);
    cin >> Key; // 输入1个查找数据
    BinSearchKey(Key);
    cout << BinSuccess << " " << BinPos << " "
         << BinCount << endl; // 最终输出结果，用于测试
  }
  return 0;
}
```



## 五、参考材料

- 严蔚敏、吴伟民，《数据结构(C语言版)》，清华大学出版社，2016. 01

