# 查找

## 定义

根据给定的某个值，在查找表中确定一个其关键字等于给定值的数据元素（或记录）。

## 分类

1）静态查找和动态查找；

注：静态或者动态都是针对查找表而言的。动态表指查找表中有删除和插入操作的表。

　　2）无序查找和有序查找。

无序查找：被查找数列有序无序均可；

有序查找：被查找数列必须为有序数列。

## 性能

平均查找长度（Average Search Length，ASL）：需和指定key进行比较的关键字的个数的期望值，称为查找算法在查找成功时的平均查找长度。

　　对于含有n个数据元素的查找表，查找成功的平均查找长度为：ASL = Pi\*Ci的和。

　　Pi：查找表中第i个数据元素的概率。

　　Ci：找到第i个数据元素时已经比较过的次数。

# 顺序表查找

## 基本思想

顺序查找也称为线形查找，属于无序查找算法。从数据结构线形表的一端开始，顺序扫描，依次将扫描到的结点关键字与给定值k相比较，若相等则表示查找成功；若扫描结束仍没有找到关键字等于k的结点，表示查找失败。

## 具体步骤

## 代码实现

## 性能分析

查找成功时的平均查找长度为：（假设每个数据元素的概率相等） ASL = 1/n(1+2+3+…+n) = (n+1)/2 ;

　　当查找不成功时，需要n+1次比较，时间复杂度为O(n);

　　所以，顺序查找的时间复杂度为O(n)。

# 有序表查找

## 折半查找

### 基本思想

也称为是折半查找，属于有序查找算法。用给定值k先与中间结点的关键字比较，中间结点把线形表分成两个子表，若相等则查找成功；若不相等，再根据k与该中间结点关键字的比较结果确定下一步查找哪个子表，这样递归进行，直到查找到或查找结束发现表中没有这样的结点。

### 具体步骤

### 代码实现

### 性能分析

## 差值查找

### 基本思想

### 具体步骤

### 代码实现

### 性能分析

## 斐波那契查找

### 基本思想

### 具体步骤

### 代码实现

### 性能分析

# 线性索引查找

## 稠密索引

## 分块索引

## 倒排索引

# 二叉排序树

## 平衡二叉树（AVL树）

## 多路查找树（B树）

# 散列表查找（哈希表）