

- 静态查找
- 动态查找
- 散列

• 查找

在一组记录集合中找到关键码值等于给定值的某个记录,或者找到关键码值符合特定条件的某些记录的过程,也称作检索。

- 查找的效率非常重要
 - **尤其对于大数据量**
 - » 需要对数据进行特殊的存储处理

提高查找效率的方法

预排序
 建立索引
 建立索引
 地域和建筑和外流域、新宿感)
 根據樣健定的值确定表中记录的位置
 缺域而提率高检索效率
 一般不允许出现重复关键码
 不适合进行范围查询

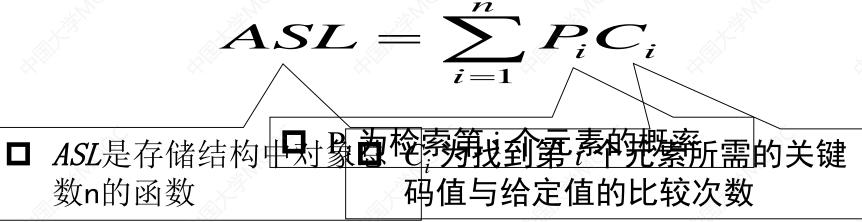
基本概念

- 查找:查询某个关键字是否在(数据元素集合)表中的过程。
 也称作检索。
- 主关键字:能够惟一区分各个不同数据元素的关键字
- 次关键字:通常不能惟一区分各个不同数据元素的关键字
- 查找成功:在数据元素集合中找到了要查找的数据元素
- 查找不成功:在数据元素集合中没有找到要查找的数据元素
- 静态查找:只查找,不改变数据元素集合内的数据元素
- 动态查找:既查找,又改变(增减)集合内的数据元素
- 静态查找表:静态查找时构造的存储结构
- 动态查找表:动态查找时构造的存储结构

查找的概念

平均检索长度 (ASL)

- 关键码的比较: 检索运算的主要操作
- 平均检索长度(Average Search Length)
 - > 检索过程中对关键码的平均比较次数
 - > 衡量检索算法优劣的时间标准



查找的概念

平均检索长度 (ASL)

- 假设线性表为(a, b, c)检索a、b、c的概率分别为0.4、0.1、0.5。
 - >顺序检索算法查找成功的平均检索长度为:

$$0.4 \times 1 + 0.1 \times 2 + 0.5 \times 3 = 2.1$$

> 即平均需要2.1次给定值与表中关键码值的比较才能找到待查元素

查找的概念

查找算法评估的几个问题

- 衡量一个查找算法还需要考虑
 - 算法所需的存储量
 - 算法的复杂性
 - *>*// . . .

- 静态查找
- 动态查找
- 散列



- 顺序查找
- 二分法查找

顺序查找

- 针对线性表里的所有记录,逐个进行关键码和给定值的比较。
 - 若某个记录的关键码和给定值比较相等,则检索成功;
 - 否则检索失败(找遍了仍找不到)。
- 存储: 可以顺序、链式
- 排序要求: 无

顺序查找

查找 Jack比较 2 次查找 Rose比较 n-1 次若查找不存在比较 n+1 次 (设置监哨岗)

顺序查找性能分析

• 查找成功 假设查找每个关键码是等概率的, $P_i = 1/n$

$$\sum_{i=0}^{n-1} P_i \cdot (n-i) = \frac{1}{n} \sum_{i=0}^{n-1} (n-i)$$
$$= \sum_{i=1}^{n} i = \frac{n+1}{2}$$

• 查找失败 假设查找失败时都需要比较n+1次(设置了一个监视哨)

顺序查找性能分析

• 平均检索长度

假设查找成功的概率为p,查找失败的概率为q=(1-p),则平均检索长度为:

ASL =
$$p \cdot \frac{n+1}{2} + q \cdot (n+1)$$

= $p \cdot \frac{n+1}{2} + (1-p)(n+1)$
= $(n+1)(1-p/2)$

$$(n+1)/2 < ASL < (n+1)$$

• 优缺点

优点:插入元素可以直接加在表尾 $\Theta(1)$

缺点: 查找时间太长 $\Theta(n)$

