

11.4散列表的性能分析

1. 平均查找长度用来度量散列表查找效率：成功、不成功
2. 关键词的比较次数，取决于产生冲突的多少，影响产生冲突的多少有以下三个因素
 1. 散列函数是否均匀
 2. 处理冲突的方法
 3. 散列表的装填因子 α

线性探测的查找性能

可以证明：线性探测法的期望探测次数满足以下公式

对插入和不成功查找而言

$$P = \frac{1}{2} \left[1 + \frac{1}{(1-\alpha)^2} \right]$$

对成功查找而言

$$P = \frac{1}{2} \left(1 + \frac{1}{1-\alpha} \right)$$

平方探测法和双散列探测法的查找性能

可以证明

对插入和不成功查找而言

$$P = \frac{1}{1-\alpha}$$

对成功查找而言

$$P = -\frac{1}{\alpha} \ln(1-\alpha)$$

分离链接法的查找性能

α 可能超过 1

期望探测次数 p

对插入和不成功查找而言

$$P = \alpha + e^{-\alpha}$$

对成功查找而言

$$P = 1 + \frac{\alpha}{2}$$

选择合适的 $h(key)$ ，散列法的查找效率期望是常数 $O(1)$ ，它几乎与关键字的空间大小 n 无关，也适合于关键字直接比较计算量大的问题

它是以较小的 α 为前提的，因此，散列方法是一个以空间换时间

散列方法的存储对关键字是随机的，不便于顺序查找关键字，也不适合于范围查找，或最大值最小值查找。

开放地址法

散列表是一个数组，存储效率高，随机查找

散列表有聚集现象

分离链接法

散列表是顺序存储和链式存储的结合，链表部分的存储效率和查找效率都比较低

关键字删除不需要“懒惰删除”法，从而没有存储垃圾

太小的 α 可能导致空间浪费，大的 α 又将付出更多的时间代价，不均匀的链表长度导致时间效率的严重下降