§3.5 Hash表

\*用链表来实现字典，每一个INSERT, DELETE或MEMBER操作，平均时间为O(n)，n为字典中元素个数。而散列(hashing)技术实现字典，以上每个操作平均时间为O(1)。

一．什么是散列技术

Hash(散列)表技术是基于建立从关键字到记录存储地址之间的函数(映射)关系而进行查找的。

设关键字的全集为U。字典存放在长为m的Hash表中。我们建立Hash函数，对每个元素的关键字k，称为k的hash函数值。Hash技术的基本思想是用作为k在hash表中的存储地址。当有两个不同的关键字和有相同的hash函数值时，即，则产生冲突。

二．链地址法

1. 方法：

链地址法是为hash表T的每一个表元建立一个链表，即为链表的表头。所有hash函数值为的元素存放在为表头的链表中。它们的插入、删除和查找均对该链表进行。

1. 链地址法的分析

设hash表长为m(m个入口)，存放了n个元素。称为装载因子。

1. 最坏情况

所有元素的hash值相同，它们存在hash表的同一个入口的链表中。最坏情况下查找需时间。

1. 平均时间

我们假设任一关键字的hash值落在T的m个入口的可能性相同。这个假设称为简单均匀散列。

定理3.1：在简单均匀散列的假设下，链地址法不成功的查找平均时间为。

证明：在简单均匀散列的假设下，任一关键字k的hash值落在m个入口的可能性相同。查找关键字k的不成功查找的平均时间是查找m条链中的一条从头到尾所需的查找时间。每条链的平均长度为，而计算h需时间，故所需总时间为。

定理3.2：在简单均匀散列的假设下，用链地址法的成功查找所需平均时间为。

证明：假设hash表中n个元素被查找的可能性相同。假设插入元素时，插入在每一条链的尾部。在成功的查找中，平均访问元素的个数是被查找的元素插入时访问的元素个数加1。插入第个元素时，每条链平均长度为。故成功查找访问元素的平均次数为

故成功查找平均时间(包括计算hash函数)为=

。

三．Hash函数

1.好的hash函数的标准

就是要满足简单均匀散列假设，即每个关键字的hash值落在m个入口的可能性相同。

在计算hash值时，我们应把每个关键字解释成一个自然数。

2.除方法

对任一关键字k， 。

如果关键字被表示成p进制数，m不宜取p的幂。m常取的值为接近2的幂的某个素数。

3.乘方法

首先用关键字k乘某个常数A，, 再取kA的小数部分乘m下取整。表达为

, 其中，表示。

假设一个机器字为w位，k占一个机器字。用k乘一个w位的整数，结果为2w位的数，取的p位有效位作为。

其中，。

一般 较好。

4.全局散列

在最坏情况下，所有关键字的散列值可能相同，查找需时间。

我们选择一个精心设计的hash函数的集合H, 然后独立于关键字随机地从H中选一个hash函数，作为我们所用的hash函数。这种方法称为全局散列。H称为是全局的，是说：对每一对不同的关键字，满足的H中的hash函数h恰有|H|/m个。其中，H, 。

从理论上可以证明，全局散列有希望得到较好的平均时间复杂性。

四．开放定址法

1. 方法：

在开放定址法中，所有元素都存在hash表中，每个表元或者保存一个元素，或者是NIL。当查找关键字k时，计算h(k)，若T[h(k)]已有元素，但不是k, 则我们按某种方式，求T中下一个位置，继续查找。下一个位置如果已有元素而且不是k, 则再求下一个位置，这个过程称为探测。在用开放定址法进行插入时，反复探测，直到找到一个空表元，插入相应的关键字。

Hash函数：

。

对每个关键字k, 探测序列

是的一个排列。

2.操作：

PROCEDURE Hash\_Insert (T, k);

BEGIN

i := 0;

REPEAT

j := h(k, i);

IF T[j] = NIL THEN

BEGIN

T[j] := k;

RETURN(j);

END

ELSE i := i+1

UNTIL i = m;

ERROR(“hash table overflow”);

END;

PROCEDURE Hash\_Search (T, k)

BEGIN

i := 0;

REPEAT

j := h(k, i);

IF T[j] = k THEN RETURN(j);

i := i+1;

UNTIL (T[j] = NIL) OR (i = m);

RETURN(NIL);

END;

当用DELETE删除一个元素时，该元素被赋予一个特殊值deleted。查找时，把deleted当作一个元素处理，而插入时，把deleted当作NIL处理。

1. 线性探测

设是通常的hash函数，，

。

1. 平方探测

，

其中，是辅助hash函数(同上), 为常数。

1. 双重散列

，

其中，和是两个辅助hash函数。

通常要求的值与hash表的长度m互素。假若不然，与m的最大公因子，则查找关键字k时，只查找了hash表的个表元。

例如：我们可以选

比稍微小一点(比如：或)。

1. 开放定址法的分析

在以下分析中，我们总是假设：对每一个关键字，它的探测序列取的每一种排列的可能性相同。这称为均匀散列假设。当k确定后，它的探测序列也就确定了。

定理3.3：在均匀散列假设下，给定具有装载因子的开放定址hash表，不成功查找的平均探测数为。

证明：在不成功查找中，除最后一次探测外，每次探测都访问一个不是预期的元素，最后一次访问一个空表元。定义

恰好次探测被占用的表元的概率，

对，，因为至多有个表元被占用。

那么平均探测数为

(3.1)

为了计算(3.1)式，定义

至少次探测被占用的表元的概率，

由概率论的知识，有

我们有

。当时，

。

推论3.4：在均匀散列的假设下，给定装载因子为的开放定址hash表，在该表中插入一个元素平均需要次探测。

证明：仅当该表中有空位时，才能插入元素，这时，。插入一个元素时，首先进行一次不成功的查找，然后在所找到的第一个空位处插入元素。故平均探测数为。

定理3.5：假设是均匀散列，并且hash表中每一个关键字被查找的可能性相同。给定装载因子为，则一次成功查找的平均探测数为

证明：查找关键字k的探测序列与插入关键字k的探测序列相同。由推论3.4，如果k是第个被插入的元素，查找k的平均探测数为。成功查找中，个关键字的平均探测数为

其中，。可以证明 。

因而有

因而本定理得证。