查　找

大连理工大学

刘 馨 月

主要内容

- 静态查找
- 动态查找
- 散列

散　列

散列

- 散列函数
- 散列冲突

散列

在随机存储中：

	学号	姓名	年龄	
01	200302	张三	19	
02	200305	李四	21	
03	200301	王五	20	

查找某一条记录需要进行一系列的“比较”。

查找的效率依赖于比较的次数。

能否在记录的关键字和存储地址之间构造这样一种关系 f，使得关键字和存储地址一一对应？

此对应关系 f 称为散列函数。

散列函数

- 把关键码值（ *key* ）映射到存储位置（ *Address* ）的函数，通常用 *Hash* 来表示：

  $$Address = Hash(key)$$

散列函数

构造散列函数时的几点要求：

- 散列函数的定义域必须包括需要存储的全部关键码，如果散列表允许有 *m* 个地址时，其值域必须在 0 到 *m*-1 之间。
- 散列函数计算出来的地址应能均匀分布在整个地址空间中：若 *key* 是从关键码集合中随机抽取的一个关键码，散列函数应能以同等概率取0 到 *m*-1 中的每一个值。
- 散列函数应是简单的，能在较短的时间内计算出结果。

散列函数

- 负载因子 $\alpha=n/m$
  - 散列表的空间大小为m
  - 填入表中的结点数为n
- 冲突
  - 某个散列函数对于不相等的关键码计算出了相同的散列地址
  - 在实际应用中，不产生冲突的散列函数极少存在
- 同义词
  - 发生冲突的两个关键码

散列函数

- 除留余数法
- 折叠法
- 平方取中法
- 基数转换法
- 直接定址法

散列函数

- 除留余数法

  H(key) = key % p

  或 H(key) = key % p + c 这里 p ≤ m；

  余数总在 0 ～ p-1 之间。

散列函数

- 示例：有一个关键码 key = 962148，散列表大小 m = 25，即 HT[25]。取质数 p= 23。散列函数 hash (key) = key % p。则散列地址为

  hash (962148) = 962148 % 23 = 12。

- 可以按计算出的地址存放记录。需要注意的是，使用上面的散列函数计算出来的地址范围是 0到 22，因此，从23到24这几个散列地址实际上在一开始是不可能用散列函数计算出来的，只可能在处理冲突时达到这些地址。

散列函数

选取 p 为质数的理由：

- 设 key 值都为奇数，选 p 为偶数；

  则 H(key) = key % p ，结果为奇数，一半单元被浪费掉。

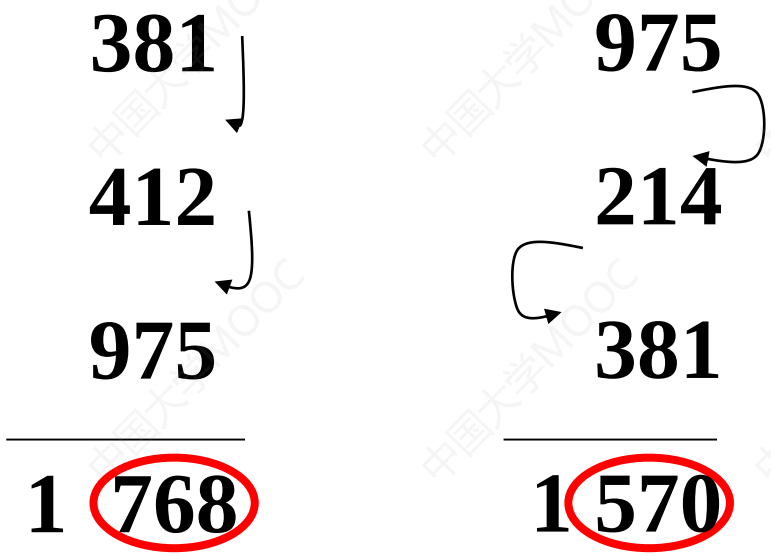
- 设 key 值都为 5 的倍数，选 p 为 95；则 H(key) = key % p ，

  结果为：0、5、10、15、…… 90 。4/5 的单元被浪费掉。

- 折叠法（移位法、分界法）

例如： key = 381412975，m=1000

选取 768 或 570 作为散列地址。



移位叠加　　边界叠加

散列函数

- 平方取中法

  例如：（4731）^{2} ＝ 223 82 361

  选取　82 （在 m ＝100 情况下）

- 此方法在词典处理中使用十分广泛。它先计算构成关键码的标识符的内码的平方，然后按照散列表的大小取中间的若干位作为散列地址。

- 设标识符可以用一个计算机字长的内码表示。因为内码平方数的中间几位一般是由标识符所有字符决定，所以即使其中有些字符相同，对不同的标识符计算出的散列地址大多不相同。

散列函数

标识符	内码	内码的平方	散列地址
A	01	01	001
*A*1	0134	20420	042
*A*9	0144	23420	342
B	02	4	004
DMAX	04150130	215264436171 00	443
*DMAX*1	0415013034	526447352215 1420	352
AMAX	01150130	135423617100	236
*AMAX*1	0115013034	345424652215 1420	652

标识符的八进制内码表示及其平方值

散列函数

- 基数转换法
  - 将关键字k转换为另外一种数字基数，再对表的大小取模。

例如：$k=(345)_{10}$ → $(423)_9$ % 表的大小

$4\times9^2+2\times9+3=345$

散列函数

- 直接定址法

H(key) = key 或

H(key) = a ×key + b

例如：key_1，key_2分别有值 10 、1000，

可以选10 、1000 作为存放地址。

散列函数

- 在实际工作中应根据关键码的特点，选用适当的方法。有人曾用“轮盘赌”的统计分析方法对它们进行了模拟分析，结论是平方取中法最接近于“随机化”。

查　找

大连理工大学

刘 馨 月