《数据结构》

散 列 表

主讲人：陈卫卫

$$\begin{cases} h_0 = hash(x) \\ h_i = (h_0 + d_i) \mod m \quad (i = 1,2,\ldots) \end{cases}$$

学习目标

1. 复述散列表的作用和意义

2. 学会散列函数的设计方法

3. 编程实现散列表的构造、插入和查找

4. 能够选择并会使用开放地址法解决散列冲突

查找学生成绩

某个班级的学习成绩单如下，请你设计一个信息组织方案，以便按照姓名查找学生成绩。

下标	name	number	chinese	math	eng	sgrade
0	张三	1201	76	85	84	
1	李四	1202	45	56	68	
2	王五	1203	93	89	96	
…	…	…	…	…	…	
n-1	牛二	1250	97	87	69	

查找学生成绩

某个班级的学习成绩单如下，请你设计一个信息组织方案，以便按照姓名查找学生成绩。

学生姓名：

张三、李四、王五、牛二、赵七、吴九

问题求解

方案一：
简单的顺序表（无序）

姓名	其他信息
张三	略
李四	略
王五	略
牛二	略
赵七	略
吴九	略

顺序查找

方案二：
按笔画排序的顺序表

姓名	其他信息
牛二	略
王五	略
吴九	略
张三	略
赵七	略
李四	略

二分查找

方案三：
检索树？

不同的组织方案下，采用了不同的查找方法，那么，自然会想到的一个问题是，查找运算的本质是什么？

查找的实质

- 查找的实质：

  完成元素值k到存储地址d的变换f，即 $d=f(k)$

- 查找算法体现了数据的逻辑特性、物理存储和操作的一致性。

查找的例子

d=*f*(姓名)

f：姓名笔画数 mod 7

d=*f*(张三)　d=3

d=*f*(李四)　d=5

d=*f*(王五)　d=1

d=*f*(牛二)　d=6

d=*f*(赵七)　d=4

d=*f*(吴九)　d=2

	姓名	其他信息
0		
1	王五	略
2	吴九	略
3	张三	略
4	赵七	略
5	李四	略
6	牛二	略
7		

散列表和散列函数

d=*f*(　　)

f：姓名笔画数 mod 7

这样一个映射称为**散列函数**。也叫**hash函数**，**杂凑函数**。

	姓名	其他信息
0		
1	王五	略
3	吴九	略
3	张三	略
4	赵七	略
5	李四	略
6	牛二	略
7		

…的散列存储空间：称作**散列表**

查找的例子

d=*f*(姓名)

f：姓名笔画数 mod 7

d=*f*(张三)　d=3

d=*f*(李四)　d=5

d=*f*(王五)　d=1

d=*f*(牛二)　d=6

d=*f*(赵七)　d=4

d=*f*(吴九)　d=2

	姓名	其他信息
0		
1	王五	略
2	吴九	略
3	张三	略
4	赵七	略
5	李四	略
6	牛二	略
7		

散列表的好处

d=*f*(姓名)

f：姓名笔画数 mod 7

显然，如果f设计的好，根据关键字，可以立刻定位到这个元素的存储位置，没有“比较”开销，算法的时间复杂度为O(1)！

	姓名	其他信息
0		
1	王五	略
2	吴九	略
3	张三	略
4	赵七	略
5	李四	略
6	牛二	略
7		

查找的例子

可是“郑甲”来了，放到哪里？

f(郑甲)=6

d=f(李四)　d=5

d=f(王五)　d=1

d=f(牛二)　d=6

d=f(赵七)　d=4

d=f(吴九)　d=2

7

1	王五	略
2	吴九	略
3	张三	略
4	赵七	略
5	李四	略
6	牛二	略
7		


PLA UNIVERSITY OF SCIENCE AND TECHNOLOGY
中国人民解放军理工大学

散列表的问题

d=*f*(姓名)

f：姓名笔画数 mod 7

d=*f*(张三)　d=3

d=*f*(吴九)　d=2

	姓名	其他信息
0		
5	李四	略
6	牛二	略
7		

可是“郑甲”来了，放到哪里？

f(郑甲)=6

散列表中的“冲突”

如果x≠y，而hash(x)=hash(y)，则称x与y发生冲突。

理想情况：若任何x≠y，都有hash(x)≠hash(y)，所设计的散列函数很均匀。查找x时，就能在a[hash(x)]中找到元素x

待解决的两个问题

如何设计散列函数hash，尽量减少冲突？

发生冲突时，如何解决冲突，提高查找效率？

散列函数的设计原则

- 根据元素取值范围和分布规律采用拼凑方式，将元素x（的值）“打乱，弄碎”，
  - 从中提取一部分信息，作为x的散列值，
  - 要尽可能均匀，不发生（或少发生）冲突

无固定的模式，只是随意拼凑

散列函数的设计方法

- 取余法
- 提取数位法
- 平方取中法
- 折叠法
- 变换基数法

（1）取余法

取余运算%

散列公式形如：hash(x)=x mod p

模数p选取不大于m的常数

0≤hash(x)≤p-1≤m-1，保证地址不越界！

通常情况下，模数p是一个接近m的最大素数时散列结果比较均匀

例如：元素个数n≤1000，散列表长度m=1024

取模数p=1019

hash(19450219)=566

hash(19761203)=755

（2）提取数位法

从元素x中取出若干位拼成整数作为x的散列值

元素取值占9 位

地址取值占3 位

一	二	三	四	五	六	七	八	九	hash(x)
9	9	0	1	2	9	1	4	6	
9	9	4	3	2	9	3	1	8	
9	9	0	4	1	9	2	7	0	
9	9	0	2	2	8	4	0	2	
9	8	4	0	1	9	9	7	5	
9	8	0	7	2	6	1	7	7	
9	8	4	5	2	0	4	9	9	

（2）提取数位法

从元素x中取出若干位拼成整数作为x的散列值

一	二	三	四	五	六	七	八	九	hash(x)
9	9	0	1	2	9	1	4	6	902
9	9	4	3	2	9	3	1	8	942
9	9	0	4	1	9	2	7	0	901
9	9	0	2	2	8	4	0	2	902
9	8	4	0	1	9	9	7	5	941
9	8	0	7	2	6	1	7	7	902
9	8	4	5	2	0	4	9	9	942

严重冲突

（2）提取数位法

从元素x中取出若干位拼成整数作为x的散列值

一	二	三	四	五	六	七	八	九	hash(x)
9	9	0	1	2	9	1	4	6	116
9	9	4	3	2	9	3	1	8	338
9	9	0	4	1	9	2	7	0	420
9	9	0	2	2	8	4	0	2	242
9	8	4	0	1	9	9	7	5	075
9	8	0	7	2	6	1	7	7	717
9	8	4	5	2	0	4	9	9	549

散列均匀


PLA UNIVERSITY OF SCIENCE AND TECHNOLOGY
中国人民解放军理工大学