



# | 冲突解决办法

## 冲突处理

- **冲突：**

是指由关键字得到的Hash地址上已有其他记录。

好的哈希函数可以减少冲突，但很难避免冲突。

- **冲突处理：**

为出现哈希地址冲突的关键字寻找下一个哈希地址。

# 冲突处理

常见的冲突处理方法有：

- 开放地址法
- 再哈希法
- 链地址法
- 公共溢出区法

# 1. 开放地址法

- 为产生冲突的地址  $H(\text{key})$  求得一个地址序列:

$$H_0, H_1, H_2, \dots, H_s \quad 1 \leq s \leq m-1$$

其中:  $H_0 = H(\text{key})$

$$H_i = (H(\text{key}) + d_i) \text{ MOD } m$$

$$i = 1, 2, \dots, s$$

其中:  $H_i$  为第 $i$ 次冲突的地址,  $i = 1, 2, \dots, s$

$H(\text{key})$  为Hash函数值

$m$  为Hash表表长

$d_i$  为增量序列

# 1. 开放地址法

对增量  $d_i$  有三种取法:

1) 线性探测再散列

$d_i = c \times i$  最简单的情况  $c=1$

2) 平方探测再散列

$d_i = 1^2, -1^2, 2^2, -2^2, \dots,$

或者  $d_i = 1^2, 2^2, 3^2, \dots$

3) 随机探测再散列

$d_i$  是一组伪随机数列

例如: 关键字集合

{ 19, 01, 23, 14, 55, 68, 11, 82, 36 }

设定哈希函数  $H(\text{key}) = \text{key} \text{ MOD } 11$  ( 表长=11 )

若采用线性探测再散列处理冲突

0	1	2	3	4	5	6	7	8	9	10
55	01	23	14	68	11	82	36	19		
1	1	2	1	3	6	2	5	1		

请求出解决冲突的查找成功的ASL  
和查找失败的ASL

例如: 关键字集合

{ 19, 01, 23, 14, 55, 68, 11, 82, 36 }

设定哈希函数  $H(\text{key}) = \text{key} \text{ MOD } 11$  ( 表长=11 )

若采用线性探测再散列处理冲突

0	1	2	3	4	5	6	7	8	9	10
55	01	23	14	68	11	82	36	19		
1	1	2	1	3	6	2	5	1		

$$\text{ASL}(\text{成功}) = (1 + 1 + 2 + 1 + 3 + 6 + 2 + 5 + 1) / 9 = 22 / 9$$

例如: 关键字集合

{ 19, 01, 23, 14, 55, 68, 11, 82, 36 }

设定哈希函数  $H(\text{key}) = \text{key} \text{ MOD } 11$  ( 表长=11 )

若采用线性探测再散列处理冲突

0	1	2	3	4	5	6	7	8	9	10
55	01	23	14	68	11	82	36	19		
1	1	2	1	3	6	2	5	1		

ASL(失败)=?

ASL(失败) =

$$(9+8+7+6+5+4+3+2+1)/11=45/11$$



例如: 关键字集合

{ 19, 01, 23, 14, 55, 68, 11, 82, 36 }

设定哈希函数  $H(\text{key}) = \text{key} \text{ MOD } 11$  ( 表长=11 )

若采用二次探测再散列处理冲突

0	1	2	3	4	5	6	7	8	9	10
55	01	23	14	11	82	68	36	19		
1	1	2	1	3	1	4	4	1		

例如: 关键字集合

{ 19, 01, 23, 14, 55, 68, 11, 82, 36 }

设定哈希函数  $H(\text{key}) = \text{key} \text{ MOD } 11$  ( 表长=11 )

若采用二次探测再散列处理冲突

0	1	2	3	4	5	6	7	8	9	10
55	01	23	14	11	82	68	36	19		
1	1	2	1	3	1	4	4	1		

**ASL(成功)=?**

**ASL(成功) =  $(1+1+2+1+3+1+4+4+1) / 9 = 2$**

例如: 关键字集合

{ 19, 01, 23, 14, 55, 68, 11, 82, 36 }

设定哈希函数  $H(\text{key}) = \text{key} \text{ MOD } 11$  ( 表长=11 )

若采用二次探测再散列处理冲突

0	1	2	3	4	5	6	7	8	9	10
55	01	23	14	11	82	68	36	19		
1	1	2	1	3	1	4	4	1		
4	3	7	6	5	4	3	2	1	0	0

**ASL(失败)=?**

例如: 关键字集合

{ 19, 01, 23, 14, 55, 68, 11, 82, 36 }

设定哈希函数  $H(\text{key}) = \text{key} \text{ MOD } 11$  ( 表长=11 )

若采用**随机数**处理冲突, 随机数假设为9

0	1	2	3	4	5	6	7	8	9	10
55	01	68	14		82	36		19	11	23
1	1	1	1		1	5		1	2	2

例如: 关键字集合

{ 19, 01, 23, 14, 55, 68, 11, 82, 36 }

设定哈希函数  $H(\text{key}) = \text{key} \text{ MOD } 11$  ( 表长=11 )

若采用**随机数**处理冲突, 随机数假设为9

0	1	2	3	4	5	6	7	8	9	10
55	01	68	14		82	36		19	11	23
1	1	1	1		1	5		1	2	2

ASL(成功)=?

$$\text{ASL(成功)} = (1+1+1+1+1+5+1+2+2) / 9 = 5/3$$

例如: 关键字集合

{ 19, 01, 23, 14, 55, 68, 11, 82, 36 }

设定哈希函数  $H(\text{key}) = \text{key} \text{ MOD } 11$  ( 表长=11 )

若采用**随机数**处理冲突, 随机数假设为9

0	1	2	3	4	5	6	7	8	9	10
55	01	68	14		82	36		19	11	23
1	1	1	1		1	5		1	2	2
3	5	4	6	1	5	2	1	3	2	4

ASL(失败)=?

ASL(失败) =  $(3+5+4+6+1+5+2+1+3+2+4) / 11 = 36 / 11$



# 1. 开放地址法

- 在 $m=16$ 的哈希表中已有关键字分别为19, 70, 33三个记录,
- 哈希函数取为 $H(key) = key \bmod 13$ ,
- 现要把第四个关键字为18的数据存入表中,
- 由哈希函数得地址为5, 产生冲突, 三种增量处理冲突。

Hash表	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
						70	19	33								
线性						70	19	33	18							
							$H_1$	$H_2$	$H_3$							
二次					18	70	19	33								
					$H_2$		$H_1$									
伪随机							70	19	33							18
							设伪随机数序列	9, 13. . . .					$H1 = 14 \bmod 16$			$=14$



## 2. 再哈希法

将n个不同哈希函数排成一个序列, 当发生冲突时, 由RHi确定第i次冲突的地址Hi。即:

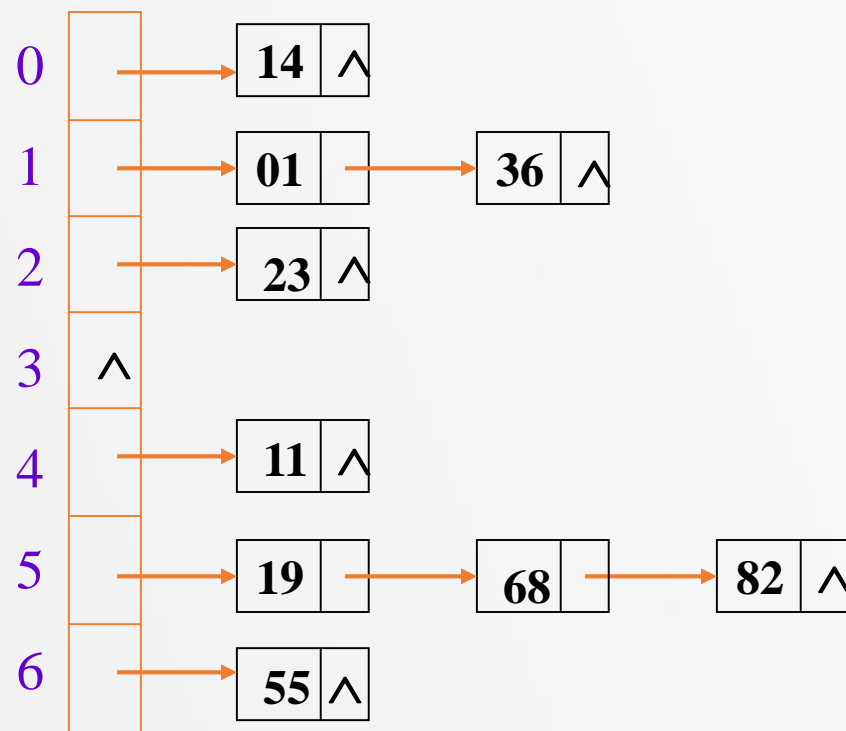
$$H_i = R H_i(\text{key}) \quad i=1, 2, \dots, n$$

其中: RHi 为不同哈希函数

这种方法不会产生“聚类”, 但会增加计算时间。

### 3. 链地址法:

将所有哈希地址相同的记录都链接在同一链表中。



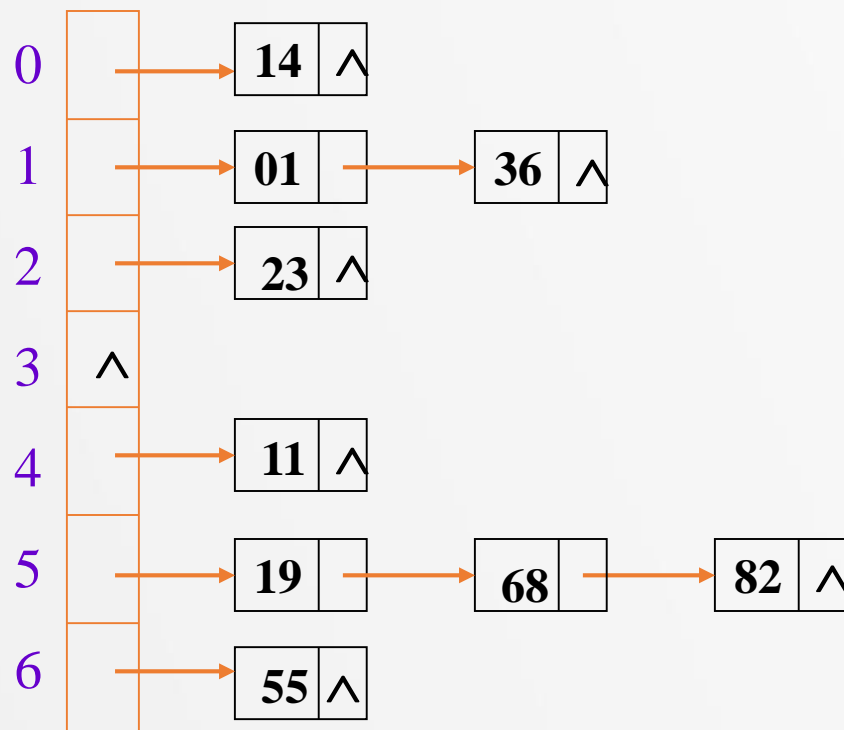
关键字集合为

{19,01,23,14,55,68,11,82,36},

哈希函数为  $H(\text{key}) = \text{key} \text{ MOD } 7$

### 3. 链地址法:

将所有哈希地址相同的记录都链接在同一链表中。



关键字集合为

{19,01,23,14,55,68,11,82,36},

哈希函数为  $H(\text{key}) = \text{key} \text{ MOD } 7$

$$\text{ASL}(\text{成功}) = (6 \times 1 + 2 \times 2 + 3) / 9 = 13 / 9$$

$$\text{ASL}(\text{失败}) = (4 \times 1 + 2 + 3) / 7 = 9 / 7$$

## 4. 公共溢出区法

- 假设某哈希函数的值域 $[0, m-1]$ ,
- 向量 $\text{HashTable}[0, m-1]$ 为**基本表**, 每个分量存放一个记录, 另设一个向量 $\text{OverTable}[0, v]$ 为**溢出表**。将与基本表中的关键字发生冲突的所有记录都填入溢出表中。
- 如一组关键字序列为 $\{19, 14, 23, 01, 68, 20, 84, 27, 55, 11, 10, 79\}$ , 哈希函数为 $H(\text{key}) = \text{key} \bmod 13$ , 采用公共溢出区法得到的结果为:

	0	1	2	3	4	5	6	7	8	9	10	11	12
hash表	^	14	^	68	^	^	19	20	^	^	23	11	^

溢出表	01	84	27	55	10	79
-----	----	----	----	----	----	----