

9.3.3 处理冲突的方法

在哈希表中，虽然冲突很难避免，但发生冲突的可能性却有大有小。这主要与三个因素有关：

- ▶ **与装填因子有关。**所谓装填因子 α 是指哈希表中已存入的元素数 n 与哈希地址空间大小 m 的比值，即 $\alpha=n/m$ 。 α 越小，冲突的可能性就越小；但 α 越小，存储空间的利用率就越低。
- ▶ **与所采用的哈希函数有关。**
- ▶ **与解决冲突的方法有关。**处理冲突就是为发生冲突的关键字的记录找到另一个“空”的哈希地址。

1. 开放定址法

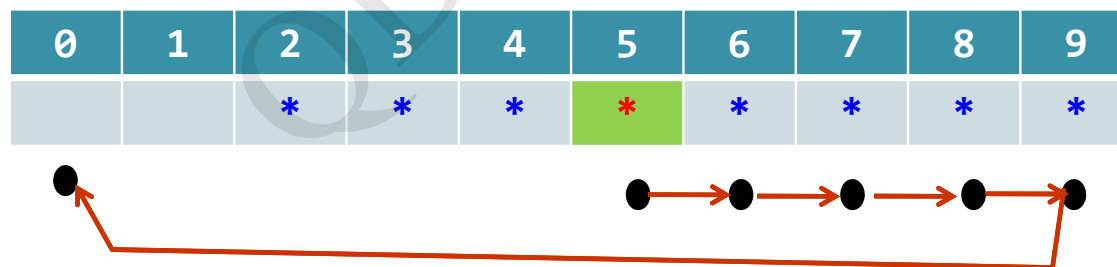
基本方法：当冲突发生时，形成某个探测序列；按此序列逐个探测散列表中的其他地址，直到找到给定的关键字或一个空地址(开放的地址)为止，将发生冲突的记录放到该地址中。散列地址的计算公式是：

$$H_i(\text{key}) = (H(\text{key}) + d_i) \text{ MOD } m, \quad i=1, 2, \dots, k (k \leq m-1)$$

- 其中： $H(\text{key})$ ：哈希函数；
- m ：散列表长度；
- d_i ：第 i 次探测时的增量序列；
- $H_i(\text{key})$ ：经第 i 次探测后得到的散列地址。

(1) 线性探测法

- 从发生冲突的地址（设为 d ）开始，依次循环探测 d 的下一个地址（当到达下标为 $m-1$ 的哈希表表尾时，下一个探测的地址是表首地址 0 ），直到找到一个空闲单元为止。
- 描述公式为： $H_i(\text{key}) = (H(\text{key}) + d_i) \text{ MOD } m$ ，其中，增量序列为： $d_i = 1, 2, 3, \dots, m-1$ 。
设 $d_0 = H(\text{key})$ ，则 $H_i(\text{key}) = d_i = (d_{i-1} + 1) \text{ mod } m \quad (1 \leq i \leq m-1)$



问题：可能出现堆积现象：

$n=6, m=10$, 关键字为 (10,11,12,19,20,21)

哈希函数： $H(\text{key}) = \text{key} \% 9$

10, 11, 12

0	1	2	3	4	5	6	7	8	9
	10	11	12						

- $19 \Rightarrow H(19)=19\%9=1$ (冲突)
- $d_0=1, d_1=(1+1)\%10=2$ (冲突)
- $d_2=(2+1)\%10=3$ (冲突)
- $d_3=(3+1)\%10=4$ (将19放在4位置)

0	1	2	3	4	5	6	7	8	9
	10	11	12	19					

- $20 \Rightarrow H(20)=20\%9=2$ (冲突)
- $d_0=2, d_1=(2+1)\%10=3$ (冲突)
- $d_2=(3+1)\%10=4$ (冲突)
- $d_3=(4+1)\%10=5$ (将20放在5位置)



哈希函数值不相同的
多个记录争夺同一个
后继哈希地址称为**二次聚集**现象。

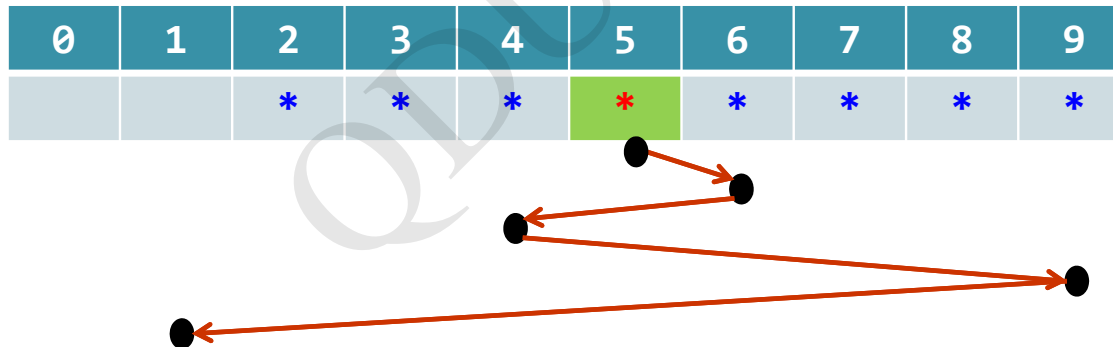
线性探测法的特点

- **优点：**只要散列表未满，**总能找到**一个不冲突的散列地址；
- **缺点：**每个产生冲突的记录被散列到离冲突最近的空地址上，从而又增加了更多的冲突机会(这种现象称为冲突的“**二次聚集**”)。

(2) 二次探测再散列

- ▶ 发生冲突时前后查找空位置。
- ▶ 描述公式为： $H_i(\text{key}) = (H(\text{key}) + d_i) \text{ MOD } m$ ，其中，增量序列为： $d_i = 1^2, -1^2, 2^2, -2^2, 3^2, \dots, \pm k^2$ ($k \leq \lfloor m/2 \rfloor$)。

设 $d_0 = H(\text{key})$, $H_i(\text{key}) = d_i = (d_0 \pm i^2) \text{ mod } m$ ($1 \leq i \leq m-1$)



- 二次探测再散列法可以避免出现堆积问题。
- 缺点是不能探测到哈希表上的所有单元，但至少能探测到一半单元。

【示例-1】假设哈希表长度 $m=13$ ，采用除留余数法加线性探测法建立如下关键字集合的哈希表：

(16,74,60,43,54,90,46,31,29,88,77)。

解： $n=11$ ， $m=13$ ，除留余数法的哈希函数为：

$$H(\text{key}) = \text{key} \bmod p$$

p 应为小于等于 m 的素数，假设 p 取值13。

哈希函数: $H(key)=key \bmod 13$

解决冲突方法: 线性探测法

- ✓ $H(16)=3$ hash[3]=16, 共1次探测
- ✓ $H(74)=9$ hash[9]=74, 共1次探测
- ✓ $H(60)=8$ hash[8]=60, 共1次探测
- ✓ $H(43)=4$ hash[4]=43, 共1次探测
- ✓ $H(54)=2$ hash[2]=54, 共1次探测
- ✓ $H(90)=12$ hash[12]=90, 共1次探测
- ✓ $H(46)=7$ hash[7]=46, 共1次探测
- ✓ $H(31)=5$ hash[5]=31, 共1次探测

- ✓ $H(29)=3$ 有冲突
- ✓ $d_0=3, d_1=(3+1) \% 13=4$ 仍有冲突
- ✓ $d_2=(4+1) \% 13=5$ 仍有冲突
- ✓ $d_3=(5+1) \% 13=6$ hash[6]=29, 共4次探测
- ✓ $H(88)=10$ hash[10]=88, 共1次探测
- ✓ $H(77)=12$ 有冲突
- ✓ $d_0=12, d_1=(12+1) \% 13=0$ hash[0]=77, 共2次探测

哈希表hash[0..12]

下标	0	1	2	3	4	5	6	7	8	9	10	11	12
k	77		54	16	43	31	29	46	60	74	88		90
探测次数	2		1	1	1	1	4	1	1	1	1		1

成功的查找：在hash中找到对应的关键字

● $H(77)=12$

$\text{hash}[12]=90 \neq 77$

$d_0=12, d_1=(12+1) \% 13=0$

$\text{hash}[0]=77$ ，共2次比较

比较的次数=探测次数



$$\text{ASL}_{\text{succ}} = \frac{1 \times 9 + 2 \times 1 + 4 \times 1}{11} = 1.364$$

哈希表hash[0..12]

下标	0	1	2	3	4	5	6	7	8	9	10	11	12
k	77		54	16	43	31	29	46	60	74	88		90

查找不成功：在hash中找不到对应的关键字x

- ◆ $H(x)=12$ $\text{hash}[12] \neq x$ $\{12, 25, 38, \dots\}$
 $d_0=12, d_1=(12+1) \% 13=0$ $\text{hash}[0] \neq x$
 $d_2=(0+1) \% 13=1$ $\text{hash}[1]$ 为空，表示查找失败，共3次比较

确定查找失败，一定比较到空为止！

哈希表hash中查找失败的所有情况的探测次数

下标	0	1	2	3	4	5	6	7	8	9	10	11	12
k	77		54	16	43	31	29	46	60	74	88		90
探测次数	2	1	10	9	8	7	6	5	4	3	2	1	3



$$ASL_{\text{unsucc}} = \frac{2+1+10+9+8+7+6+5+4+3+2+1+3}{13} = 4.692$$

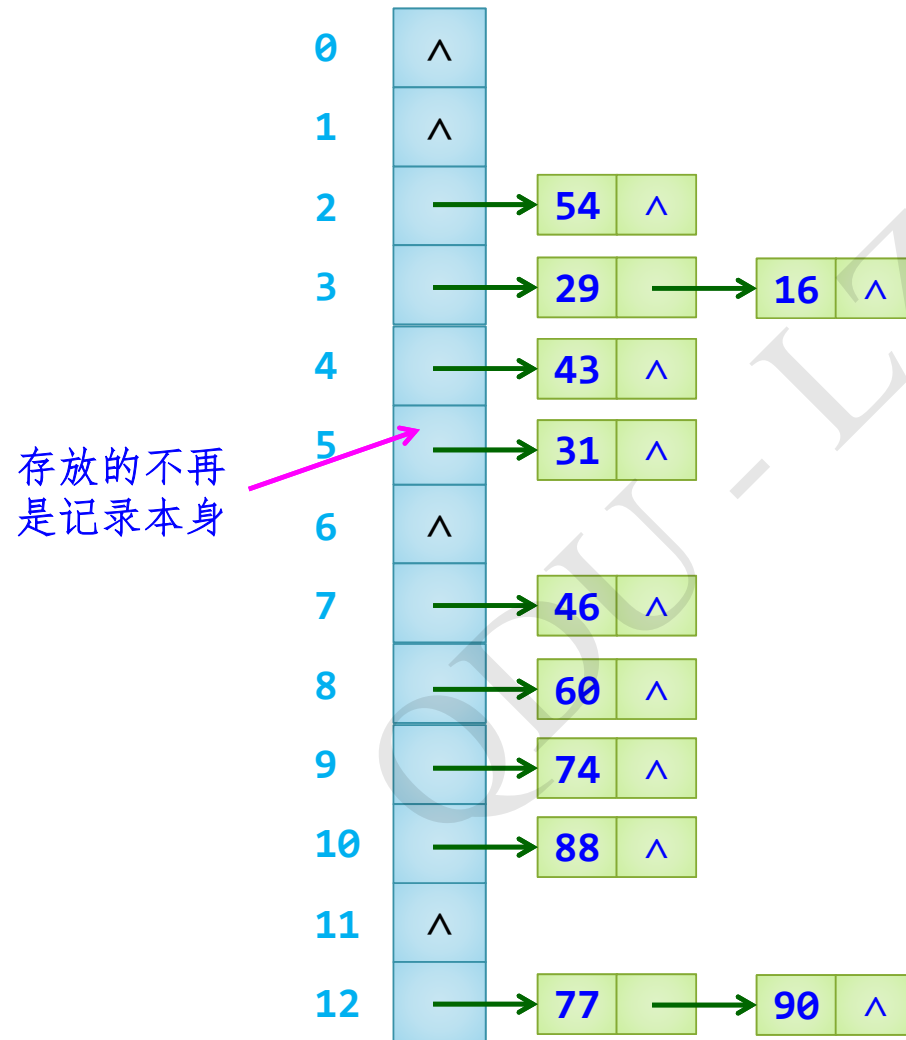
2. 链地址法

- ▶ **链地址法**：将所有关键字为**同义词**(哈希地址相同)的记录存储在一个单链表中，并用一维数组存放链表的头指针。
- ▶ 在这种方法中，哈希表每个单元中存放的不再是记录本身，而是相应同义词单链表的头指针。
- ▶ 由于单链表中可插入任意多个结点，所以此时**装填因子 α** 根据同义词的多少既可以设定为大于1，也可以设定为小于或等于1，**通常取 $\alpha=1$** 。

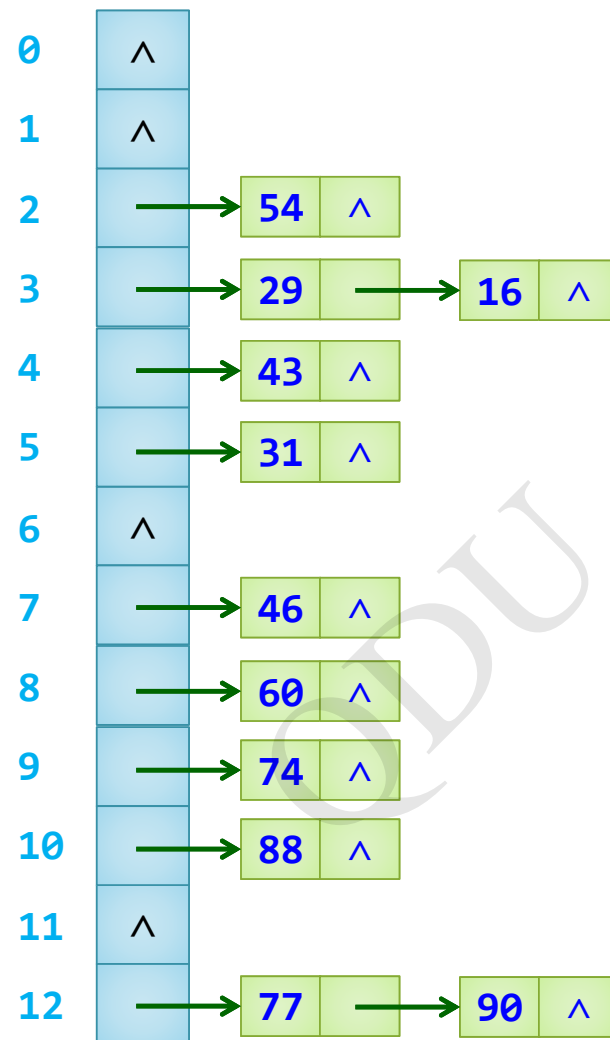
【示例-2】假设哈希表长度 $m=13$ ，采用采用除留余数法加链地址法建立如下关键字集合的哈希表：

(16, 74, 60, 43, 54, 90, 46, 31, 29, 88, 77)。

解：采用链地址法解决冲突建立的链表如下图所示。



成功的查找： 在hash中找到对应的关键字



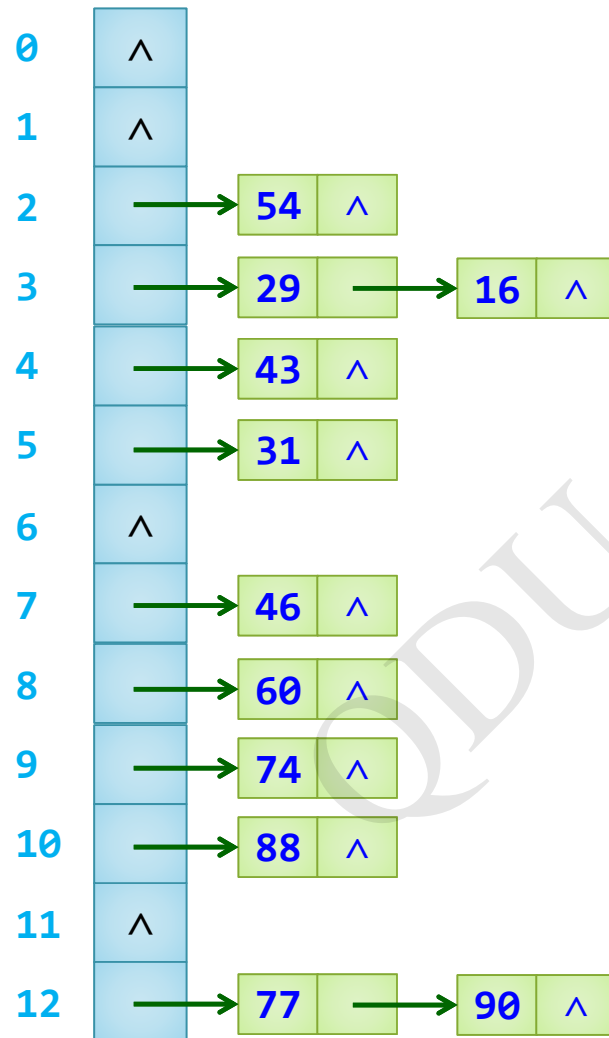
● $H(77)=12$, 在hash[12]的单链表中
中共1次比较

比较的次数=对应结点在
单链表中的序号



$$ASL_{succ} = \frac{1*9 + 2*2}{11} = 1.18$$

不成功的查找： 在hash中找不到对应的关键字x



● $H(x)=3$, 在hash[3]的单链表中

● 共2次比较

比较的次数=对应单链表中的结点个数



$$ASL_{\text{unsucc}} = \frac{1*7 + 2*2}{13} = 0.846$$

【示例-3】 将关键字序列(7,8,30,11,18,9,14)散列存储到散列表中，散列表的存储空间是一个下标从0开始的一维数组，散列函数为： $H(key)=(key \times 3) \bmod 7$ ，处理冲突采用线性探测再散列法，要求装填（载）因子为0.7。

(1) 请画出所构造的散列表。

(2) 分别计算等概率情况下，查找成功和查找不成功的平均查找长度。

说明：本题为2010年全国考研题。

解：（1） $n=7$ ， $\alpha=0.7=n/m$ ，则 $m=n/0.7=10$ 。

计算各关键字存储地址的过程如下：

$$H(7)=7 \times 3 \bmod 7=0$$

$$H(8)=8 \times 3 \bmod 7=3$$

$$H(30)=30 \times 3 \bmod 7=6$$

$$H(11)=11 \times 3 \bmod 7=5$$

$$H(18)=18 \times 3 \bmod 7=5 \quad \text{冲突}$$

$$d_1=(5+1) \bmod 10=6 \quad \text{仍冲突}$$

$$d_2=(6+1) \bmod 10=7$$

$$H(9)=9 \times 3 \bmod 7=6 \quad \text{冲突}$$

$$d_1=(6+1) \bmod 10=7 \quad \text{仍冲突}$$

$$d_2=(7+1) \bmod 10=8$$

$$H(14)=14 \times 3 \bmod 7=0 \quad \text{冲突}$$

$$d_1=(0+1) \bmod 10=1$$

构造的哈希表：

下标	0	1	2	3	4	5	6	7	8	9
关键字	7	14		8		11	30	18	9	
探测次数	1	2		1		1	1	3	3	

(2) 在等概率情况下：

$$ASL_{成功} = (1+2+1+1+1+3+3)/7 = 12/7 = 1.71$$

不成功的情况下所有探测次数:

下标	0	1	2	3	4	5	6	7	8	9
关键字	7	14		8		11	30	18	9	
探测次数	3	2	1	2	1	5	4	3	2	1

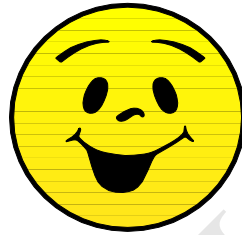
所以有:

$$ASL_{\text{不成功}} = (3+2+1+2+1+5+4)/7 = 18/7 = 2.57$$

- n 个关键字的构造顺序不同得到的哈希表不同
- 平均查找长度ASL也不同
- 考虑所有顺序构造哈希表的平均情况

平均情况下的平均查找长度：

解决冲突的方法	平均查找长度ASL	
	成功的查找	不成功的查找
线性探测法	$\frac{1}{2}(1 + \frac{1}{1-\alpha})$	$\frac{1}{2}(1 + \frac{1}{(1-\alpha)^2})$
平方探测法	$-\frac{1}{\alpha} \log_e(1-\alpha)$	$\frac{1}{1-\alpha}$
链地址法	$1 + \frac{\alpha}{2}$	$\alpha + e^{-\alpha} \approx \alpha$



— END —