

9.4 哈希表的查找

9.4.1 哈希表的基本概念

1、哈希表适合情况



示例

学号	姓名
201001001	张三
201001003	李四
...	
201001025	王五

记录数 $n=20$, 无序



传统存储方法: 存放在一个数组中

0	1	...	19
201001001	201001003	...	201001025
张三	李四	...	王五

} 20个元素空间

查找学号为**201001025**的学生姓名:

- 从头到尾顺序查找, 时间复杂度为 $O(n)$ 。
- 若学号有序, 二分查找, 时间复杂度为 $O(\log_2 n)$ 。

另一种存储结构:

学号	姓名
201001001	张三
201001003	李四
...	
201001025	王五

$n=20, m=30$

存放地址=学号-201001001

0	1	2	...	24	...	29
201001001	空闲	201001003	...	201001025	...	空闲
张三	空闲	李四	...	王五	...	空闲

哈希表

30个元素空间

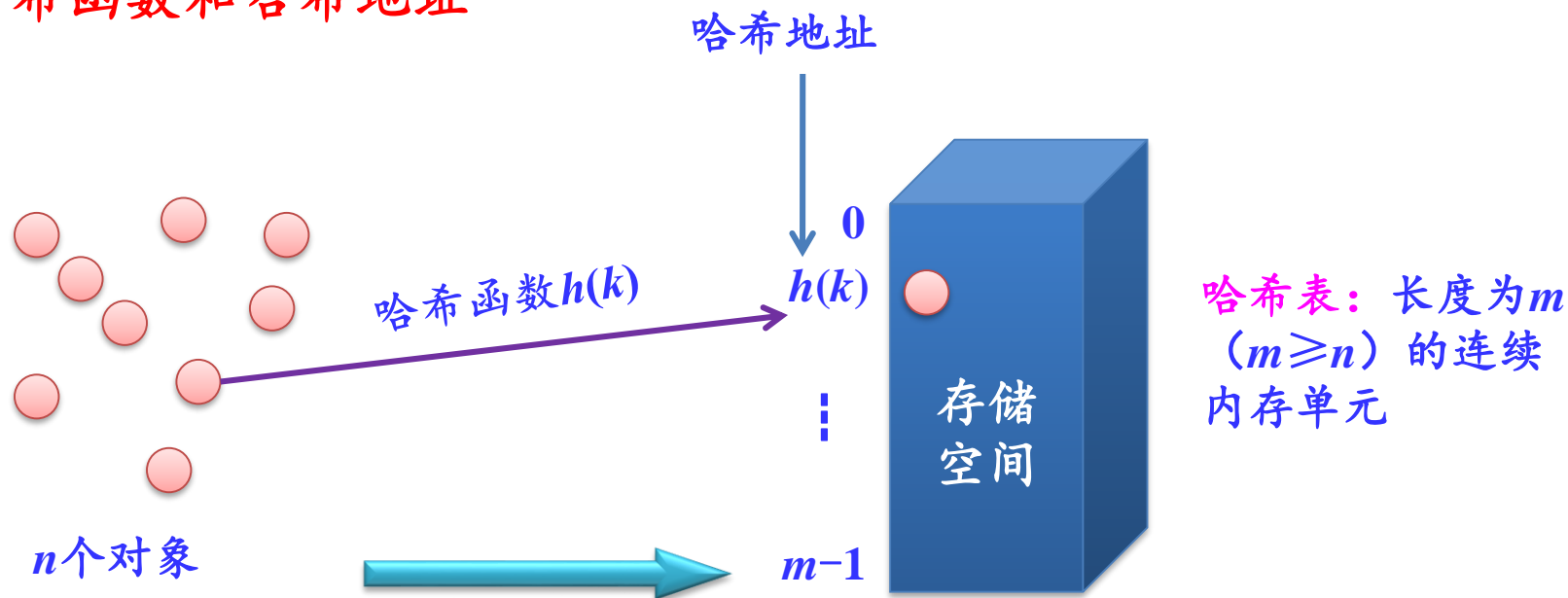
查找学号为201001025的学生姓名:

- ① 计算:地址 $d=201001025-201001001=24$
- ② 和24处的学号比较, 相等, 返回姓名“王五”

时间复杂度 $O(1)$

2、几个概念

① 哈希函数和哈希地址



哈希函数: 把关键字为 k_i 的对象存放在相应的哈希地址中

② 哈希冲突

对于两个关键字分别为 k_i 和 k_j ($i \neq j$) 的记录, 有 $k_i \neq k_j$, 但 $h(k_i) = h(k_j)$ 。
把这种现象叫做**哈希冲突**（**同义词冲突**）。

在哈希表存储结构的存储中, 哈希冲突是很难避免的!!!

3、哈希表设计

哈希表设计主要需要解决哈希冲突。实际中哈希冲突是难以避免的，主要与3个因素有关：

- 与装填因子有关。装填因子 α =存储的记录个数/哈希表的大小
 $=n/m \Rightarrow \alpha$ 越小，冲突的可能性就越小； α 越大（最大可取1），冲突的可能性就越大。通常使最终的控制控制在0.6~0.9的范围内。
- 与所采用的哈希函数有关。好的哈希函数会减少冲突的发生；不好的哈希函数会增加冲突的发生。
- 与解决冲突方法有关。好的哈希冲突解决方法会减少冲突的发生。

所以哈希表设计的重点：

- 尽可能设计好的哈希函数
- 设计解决冲突的方法。

思考题

好的哈希函数应该具有什么特点？

9.4.2 哈希函数构造方法

1、直接定址法

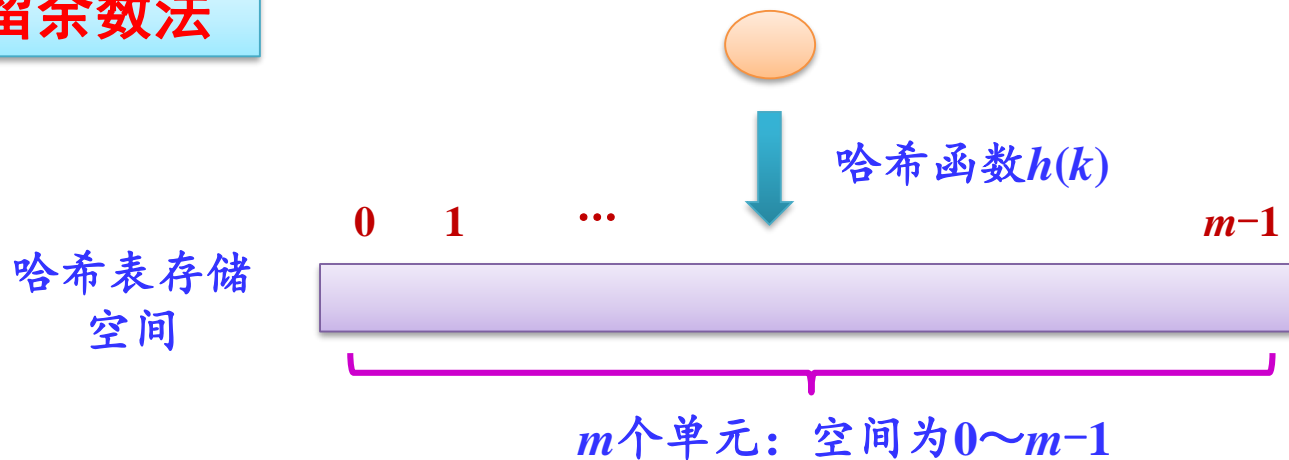
直接定址法是以关键字 k 本身或关键字加上某个数值常量 c 作为哈希地址的方法。直接定址法的哈希函数 $h(k)$ 为：

$$h(k) = k + c$$

例如：

$$h(\text{学号}) = \text{学号} - 201001001$$

2、除留余数法



除留余数法的哈希函数 $h(k)$ 为：

$$h(k)=k \bmod p \quad (\text{mod 为求余运算, } p \leq m)$$

p 最好是质数（素数）。

3、数字分析法

关键字

9	2	3	1	7	6	0	2
9	2	3	2	6	8	7	5
9	2	7	3	9	6	2	8
9	2	3	4	3	6	3	4
9	2	7	0	6	8	1	6
9	2	7	7	4	6	3	8
9	2	3	8	1	2	6	2
9	2	3	9	4	2	2	0

为两位取最后
地址哈希作后



0	2
7	5
2	8
3	4
1	6
3	8
6	2
2	0



哈希地址的集合为{2,75,28,34,16,38,62,20}。

大数值范围

哈希函数 $h(k)$

小数值范围

【例9-6】 假设哈希表长度 $m=13$ ，采用除留余数法哈希函数建立如下关键字集合的哈希表： $\{16, 74, 60, 43, 54, 90, 46, 31, 29, 88, 77\}$ ，共11个关键字。

解： $n=11$ ， $m=13$ ，设计除留余数法的哈希函数为：

$$h(k)=k \bmod p$$

p 应为小于等于 m 的素数，设 $p=13$ 。

关键字: 16 74 60 43 54 90 46 31 29 88 77

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

$$h(29)=3$$

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

注意: 存在哈希冲突。

9.4.3 哈希冲突解决方法

1、开放定址法

开放定址法：冲突时找一个新的空闲的哈希地址。



怎么找空闲单元？

实例：晚到电影院找座位的情况就是采用开放定址法。

(1) 线性探查法

线性探查法的数学递推描述公式为：

$$d_0 = h(k)$$

$$d_i = (d_{i-1} + 1) \bmod m \quad (1 \leq i \leq m-1)$$

思路：在电影院中找被占用位置的后面空位置！模 m 是为了保证找到的位置在 $0 \sim m-1$ 的有效空间中。

非同义词冲突：哈希函数值不相同的两个记录争夺同一个后继哈希地址 \Rightarrow 堆积（或聚集）现象。

(2) 平方探查法

平方探查法的数学描述公式为：

$$d_0 = h(k)$$

$$d_i = (d_0 \pm i^2) \bmod m \quad (1 \leq i \leq m-1)$$

查找的位置依次为： d_0 、 d_0+1 、 d_0-1 、 d_0+4 、 d_0-4 、...

思路：在电影院中找被占用位置的前后空位置！

平方探查法是一种较好的处理冲突的方法，可以避免出现堆积现象。它的缺点是不能探查到哈希表上的所有单元，但至少能探查到一半单元。

【例9-7】 假设哈希表长度 $m=13$ ，采用除留余数法哈希函数建立如下关键字集合的哈希表：

{16, 74, 60, 43, 54, 90, 46, 31, 29, 88, 77}。

并采用线性探查法解决冲突。

关键字：16 74 60 43 54 90 46 31 29 88 77



$$h(29)=3$$

⇒ 冲突

$$d_0=3, d_1=(3+1) \% 13=4$$

⇒ 仍冲突

$$d_2=(4+1) \% 13=5$$

⇒ 仍冲突

$$d_3=(5+1) \% 13=6$$

⇒ OK

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

共探查4次

关键字：16 74 60 43 54 90 46 31 29 88 77

$h(88)=10$

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

共探查1次

关键字: 16 74 60 43 54 90 46 31 29 88 77
↑

$h(77)=12$ \Rightarrow 冲突

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

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

共探查2次

哈希表创建完毕

最终的哈希表

哈希表ha[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
探查次数	2		1	1	1	1	4	1	1	1	1		1



哈希表的构成：

- 哈希函数：本例为 $h(k)=k \bmod 13$
- 解决冲突方法：本例为线性探查法

开放定址法哈希表查找 k 过程:

```
d=h(k);
```

```
while (ha[d]!=空 && ha[d]!=k)
```

```
    d=采用某种探查法求出下一地址;
```

```
if (ha[d]==空)
```

```
    return 失败标记;
```

```
else
```

```
    return ha[d];
```

成功查找的情况

查找关键字为**29**的记录：

$$h(29)=29\%13=3 : 16 \neq \mathbf{29} ;$$

$$d_0=3, d_1=(3+1)=4 : 43 \neq \mathbf{29} ;$$

$$d_2=(4+1)=5 : 31 \neq \mathbf{29} ;$$

$$d_3=(5+1)=6 : 29 = \mathbf{29}。 \text{成功！}$$

需要**4**次关键字比较

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
探查次数	2		1	1	1	1	4	1	1	1	1		1

哈希表成功查找完毕

① 对于前面构建的哈希表：成功查找ASL计算

哈希表ha[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
探查次数	2		1	1	1	1	4	1	1	1	1		1

探查次数恰好等于查找到该记录所需要的关键字比较次数！



$$ASL_{\text{成功}} = \frac{2+1+1+1+1+4+1+1+1+1+1}{11} = 1.364$$

不成功查找的情况

查找关键字 $x=47$ 的记录

$h(47)=47\%13=8:$ $\Rightarrow 60 \neq 47$

$d_0=8, d_1=(8+1)=9:$ $\Rightarrow 74 \neq 47$

$d_2=(9+1)=10:$ $\Rightarrow 88 \neq 47$

$d_3=(10+1)=11:$ 此处为空，查找失败！

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
探查次数	2		1	1	1	1	4	1	1	1	1		1

需要4次关键字比较

哈希表失败查找完毕

② 对于前面构建的哈希表：不成功查找ASL计算

哈希表ha[0..12]

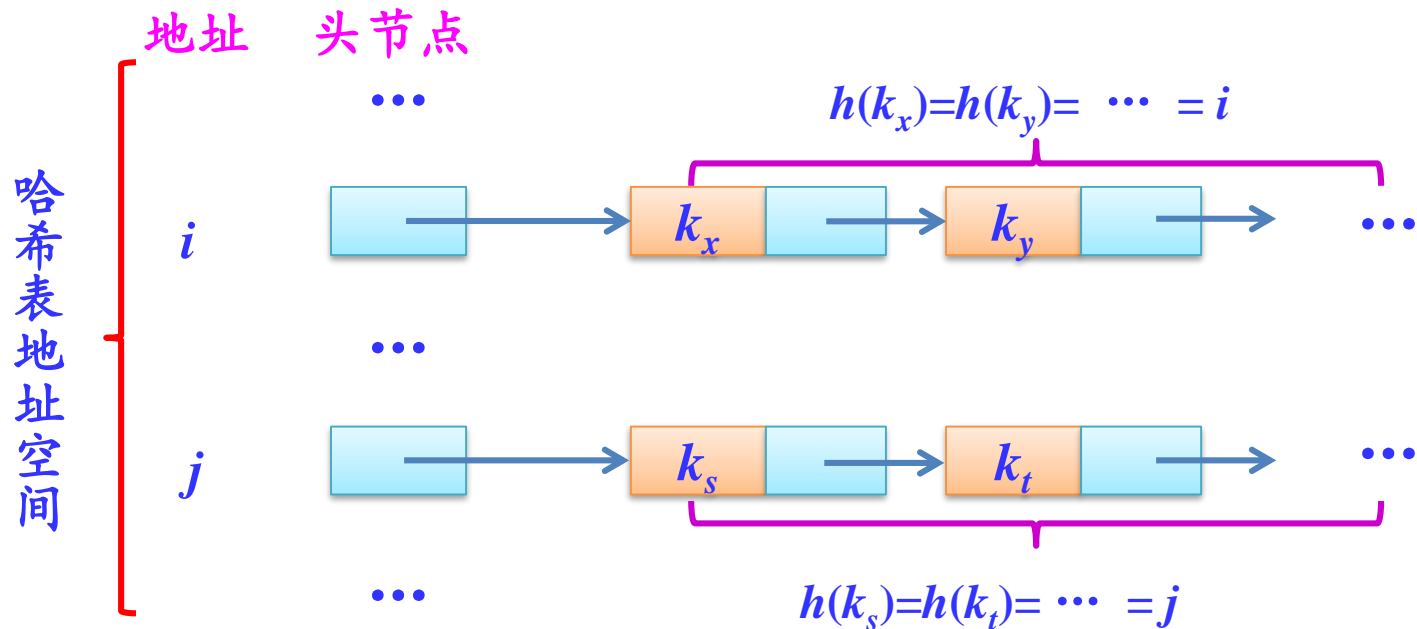
下标	0	1	2	3	4	5	6	7	8	9	10	11	12
k	77		54	16	43	31	29	46	60	74	88		90
探查次数	2	1	10	9	8	7	6	5	4	3	2	1	3

一个关键字不在关键字集合中，其哈希函数值为0，线性探查法找到空位置，需要10次关键字比较

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

2、拉链法

拉链法是把所有的同义词用单链表链接起来的方法。



【例9-8】 对例9-7的关键字序列，构造采用**拉链法**解决冲突的哈希表。

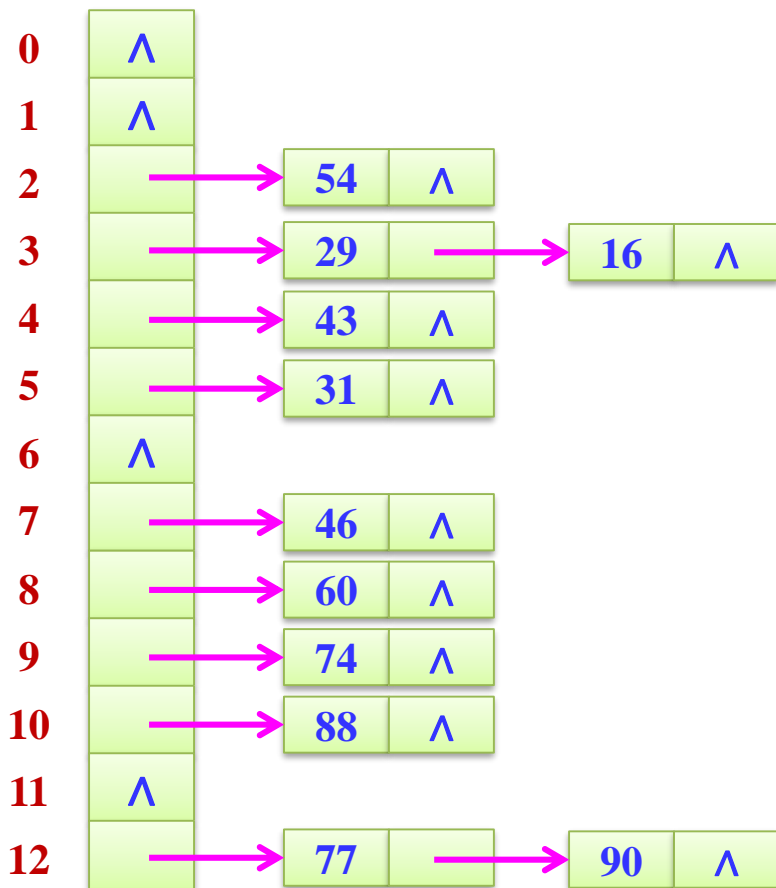
关键字集合：

{16,74,60,43,

54,90,46,31,2

9,88,77}

采用
拉链法
构造的
哈希表
ha[0..12]



拉链法哈希表查找 k 过程:

```
d=h(k);
```

```
p=ha[d];
```

```
while (p!=NULL && p->key!=k)
```

```
    p=p->next; //在ha[d]的单链表中查找
```

```
if (p==NULL)
```

```
    return 失败标记;
```

```
else
```

```
    return p所指节点;
```

成功查找的情况

查找关键字为**16**的记录:

$$h(16)=16\%13=3$$

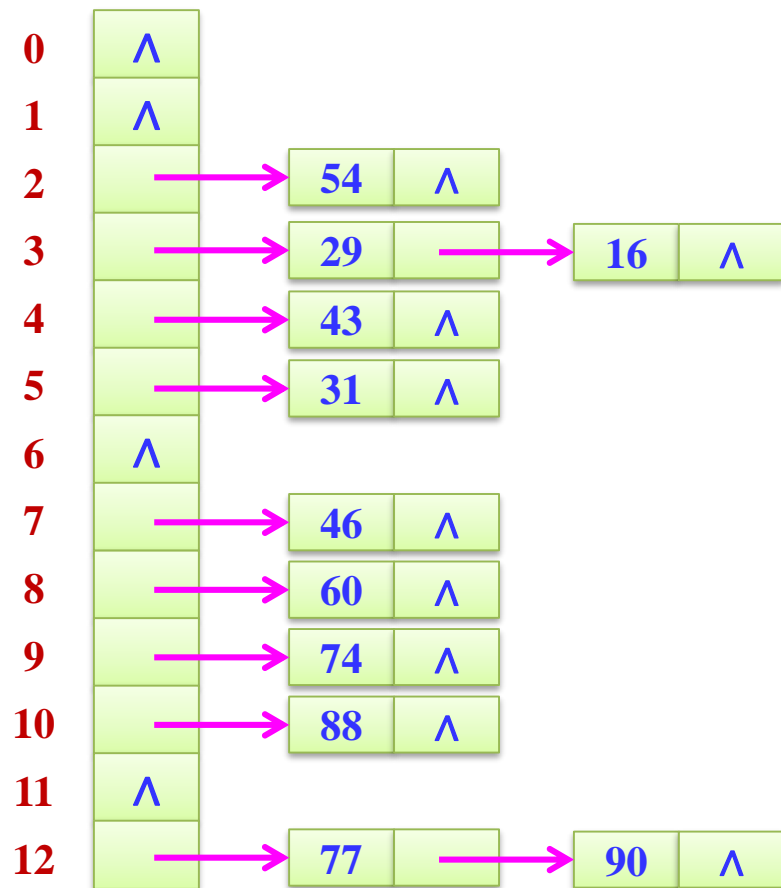
p指向ha[3]的第1个节点, $29 \neq 16$;

p指向ha[3]的第2个节点, $16 = 16$ 。

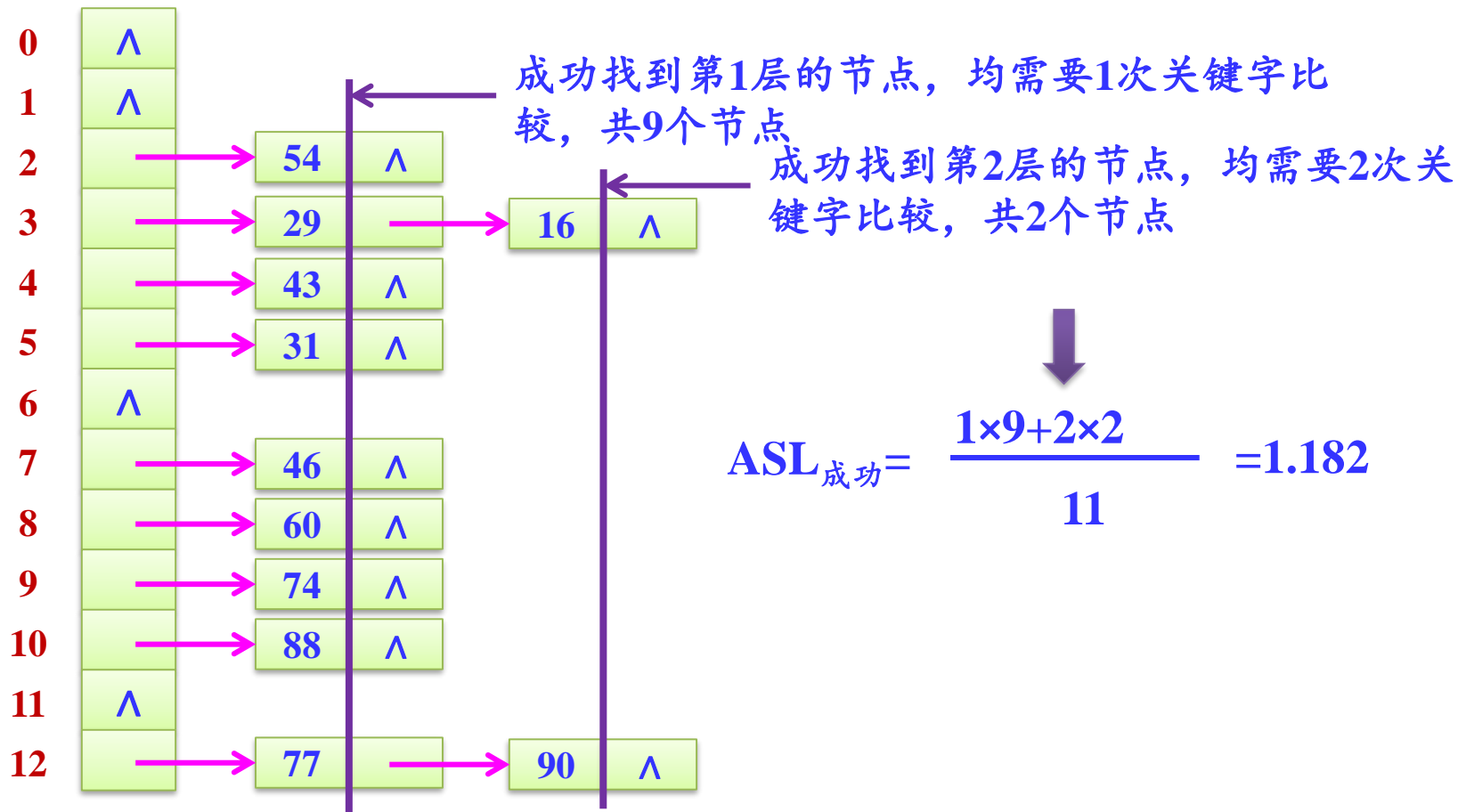
成功!

2次关键字比较

哈希表成功查找完毕



① 拉链法中成功查找的ASL计算



不成功查找的情况

查找关键字为47的记录:

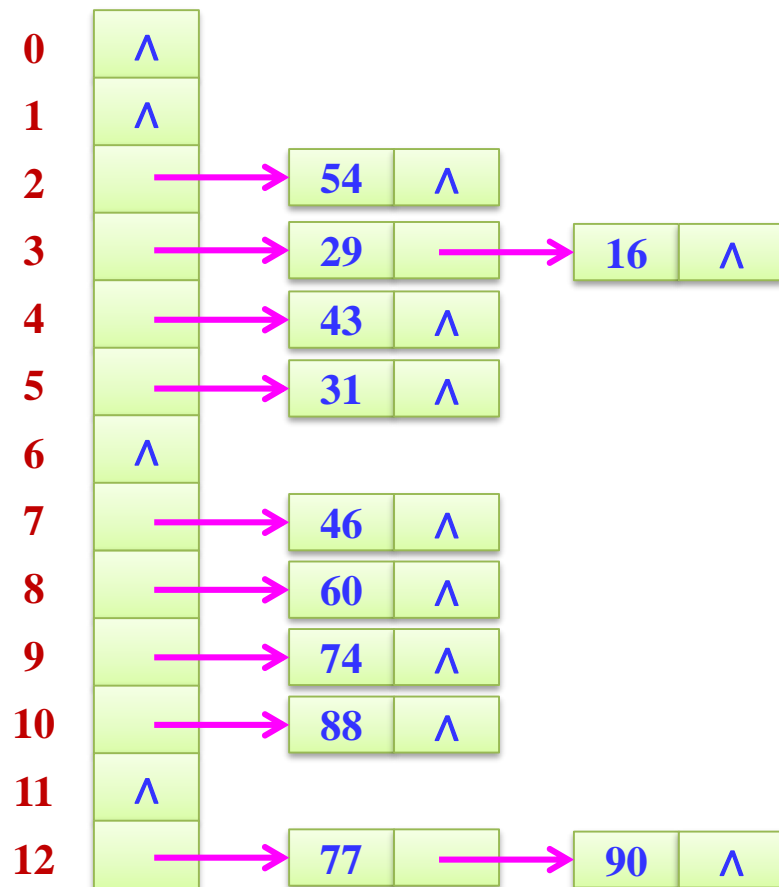
$$h(47)=47\%13=8$$

p指向ha[8]的第1个节点, $60 \neq 47$;

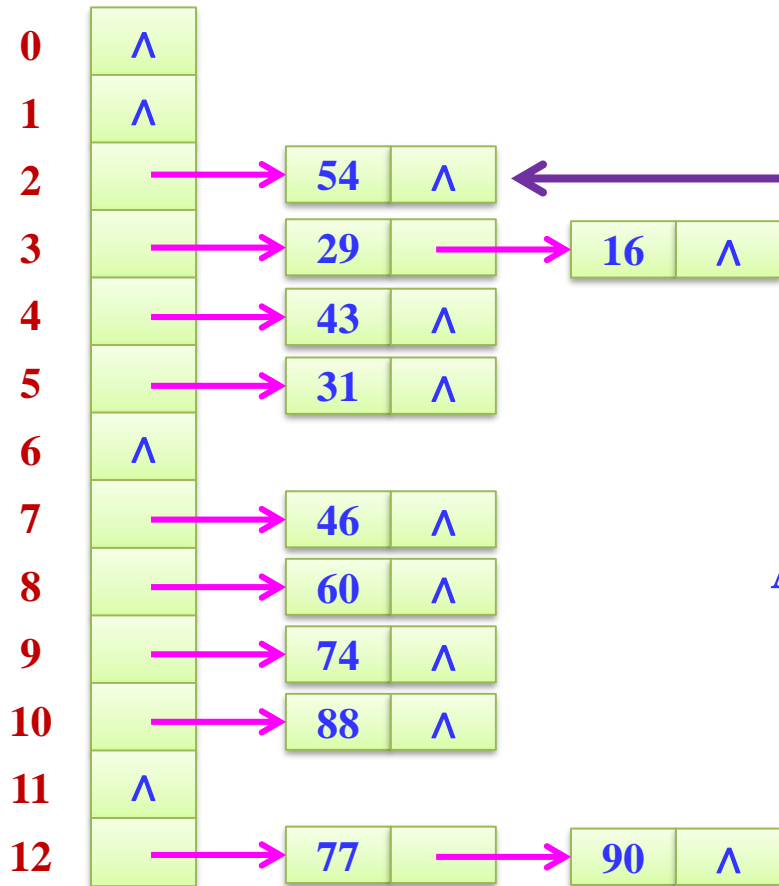
p=NULL。失败!

1次关键字比较

哈希表不成功查找完毕



② 拉链法不成功查找的ASL计算



有1个节点的单链表，不成功查找需要1次关键字比较，共有7个这样的单链表
有2个节点的单链表，不成功查找需要2次关键字比较，共有2个这样的单链表

$$ASL_{\text{不成功}} = \frac{1 \times 7 + 2 \times 2}{13} = 0.846$$

$\alpha = n/m$

思考题

开放定址法和拉链法各有什么优缺点？

图片编号: 1022666
来源: WWW.OOOPIC.COM



——本章完——