

---

# 数据结构

## 哈希表

---

创客学院 小美老师

# Hash表的查找

理想的查找方法是：对给定的k，不经任何比较便能获取所需的记录，其查找的时间复杂度为常数级 $O(1)$ 。

这就要求在建立记录表的时候，确定记录的key与其存储地址之间的关系f，即使key与记录的存放地址H相对应：

key  $\xrightarrow{f}$  H: 记 录

# Hash表的查找

当要查找 $\text{key} = k$ 的记录时，通过关系 $f$ 就可得到相应记录的地址而获取记录，从而免去了 $\text{key}$ 的比较过程。

这个关系 $f$ 就是所谓的Hash函数（或称散列函数、杂凑函数），记为 $H(\text{key})$ 。

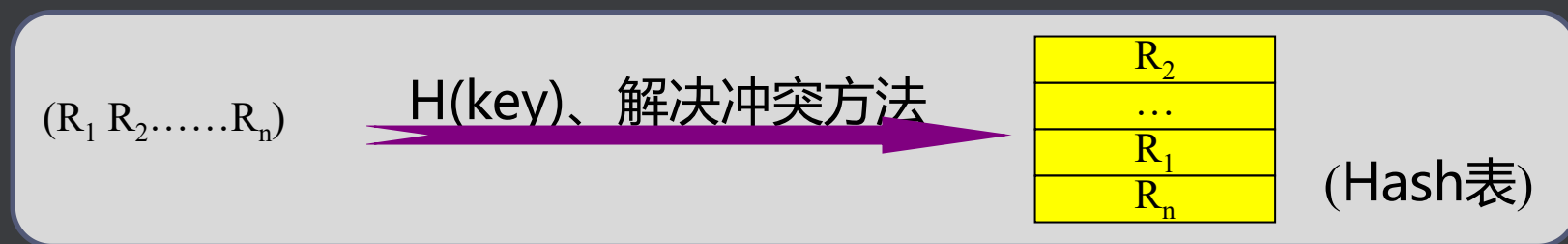
它实际上是一个地址映象函数，其自变量为记录的 $\text{key}$ ，函数值为记录的存储地址（或称Hash地址）。

# Hash表的查找

- 不同的key可能得到同一个Hash地址，即当 $key_1 \neq key_2$ 时，可能有 $H(key_1) = H(key_2)$ ，此时称 $key_1$ 和 $key_2$ 为**同义词**。这种现象称为“**冲突**”或“**碰撞**”，因为一个数据单位只可存放一条记录。
- 一般，选取Hash函数只能做到使冲突尽可能少，却不能完全避免。这就要求在出现冲突之后，寻求适当的方法来解决冲突记录的存放问题。

# Hash表的查找

根据选取的Hash函数 $H(\text{key})$ 和处理冲突的方法，将一组记录 $(R_1 R_2 \dots R_n)$ 映射到记录的存储空间，所得到的记录表称为Hash表，如图：



# Hash表的查找

选取（或构造）Hash函数的方法很多，原则是尽可能将记录均匀分布，以减少冲突现象的发生。以下介绍几种常用的构造方法。

直接地址法

平方取中法

叠加法

保留除数法

随机函数法

# 保留除数法

又称质数除余法，设Hash表空间长度为m，选取一个不大于m的最大质数p，

令： $H(\text{key}) = \text{key} \% p$

# 保留除数法

设记录的key集合 $k=\{28, 35, 63, 77, 105, \dots\}$ , 若选取 $p=21=3*7$  (包括质数因子7), 有:

key: 28 35 63 77 105 .....

$H(\text{key})=\text{key}\%21$ : 7 14 0 14 0 .....

使得包含质数因子7的key都可能被映射到相同的单元, 冲突现象严重。



# 保留除数法

若取 $p=19$ （质数），同样对上面给定的key集合 $k$ ，有：

key: 28 35 63 77 105

$H(\text{key})=\text{key}\%19$ : 9 16 6 1 10

$H(\text{key})$ 的随机度就好多了。

# 处理冲突的方法

选取随机度好的Hash函数可使冲突减少，一般来讲不能完全避免冲突。设Hash表地址空间为 $0 \sim m-1$ （表长为 $m$ ）：



冲突是指：表中某地址 $j \in [0, m-1]$ 中已存放有记录，而另一个记录的 $H(key)$ 值也为 $j$ 。

# 处理冲突的方法

- 处理冲突的方法一般为：在地址j的前面或后面找一个空闲单元存放冲突的记录，或将相冲突的诸记录拉成链表。
- 在处理冲突的过程中，可能发生一连串的冲突现象，即可能得到一个地址序列 $H_1, H_2, \dots, H_n$ ， $H_i \in [0, m-1]$ 。 $H_1$ 是冲突时选取的下一地址，而 $H_1$ 中可能已有记录，又设法得到下一地址 $H_2, \dots$ 直到某个 $H_n$ 不发生冲突为止。这种现象称为“**聚积**”，它严重影响了Hash表的查找效率。

# 处理冲突的方法

- 冲突现象的发生有时并不完全是由于Hash函数的随机性不好引起的，聚集的发生也会加重冲突。
- 还有一个因素是表的装填因子 $\alpha$ ， $\alpha = n/m$ ，其中 $m$ 为表长， $n$ 为表中记录个数。一般 $\alpha$ 在0.7 ~ 0.8之间，使表保持一定的空闲余量，以减少冲突和聚集现象。

# 开放地址法

当发生冲突时，在 $H(\text{key})$ 的前后找一个空闲单元来存放冲突的记录，即在 $H(\text{key})$ 的基础上获取下一地址：

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

其中 $m$ 为表长， $\%$ 运算是保证 $H_i$ 落在 $[0, m-1]$ 区间；

$d_i$ 为地址增量。 $d_i$ 的取法有多种：

- (1)  $d_i = 1, 2, 3, \dots, (m-1)$ ——称为线性探查法；
- (2)  $d_i = 1^2, -1^2, 2^2, -2^2, \dots$ ——称为二次探查法。

# 开放地址法

设记录的key集合 $k=\{23, 34, 14, 38, 46, 16, 68, 15, 07, 31, 26\}$ ,  
记录数 $n=11$ 。

令装填因子 $\alpha=0.75$ , 取表长 $m=\lceil n/\alpha \rceil = 15$ 。

用“保留余数法”选取Hash函数 ( $p=13$ ) :

$$H(\text{key}) = \text{key} \% 13$$

# 开放地址法

$$H(\text{key}) = \text{key} \% 13$$

$k = \{23, 34, 14, 38, 46, 16, 68, 15, 07, 31, 26\}$

$$H(\text{key}) = \text{key} \% 13; \quad H_i = (H(\text{key}) + d_i) \% 15; \quad d_i = 1, 2, 3, \dots, (m-1)$$

HT:	26	14	15	16	68	31	^	46	34	07	23	^	38	^	^
H(key)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

# 链地址法

发生冲突时，将各冲突记录链在一起，即同义词的记录存于同一链表。

设 $H(\text{key})$ 取值范围（值域）为 $[0, m-1]$ ,

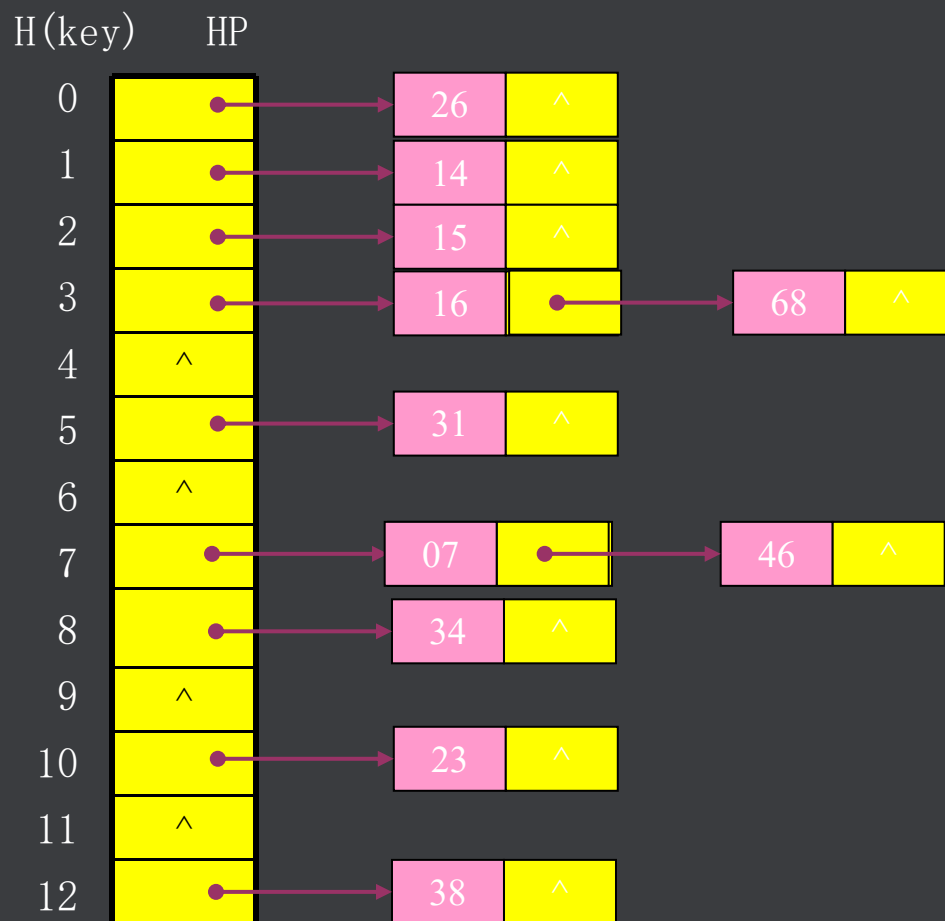
建立头指针向量 $HP[m]$ ,

$HP[i]$  ( $0 \leq i \leq m-1$ ) 初值为空。

设 $H(\text{key}) = \text{key} \% 13$

$k = \{ 23, 34, 14, 38, 46, 16, 68, 15, 07, 31, 26 \}$

链地址法解决冲突的优点：无聚积现象；删除表中记录容易实现。





扫一扫，获取更多信息



THANK YOU