

# 散列表

字典：放入、查找、删除。

符号表：键  $\rightarrow$  值

编译器：变量名  $\rightarrow$  存储地址

最坏  $O(n)$   $\rightarrow$  期望  $O(1)$

散列 + 集合

数组：下标 换为地址  $O(1)$  时间，推广这种思想

直接存取 容量问题  $\rightarrow$  散列表  $\rightarrow$  以函数算出键的下标  $\rightarrow$  进而获取值

问题：碰撞  $\rightarrow$  相同下标

不同的键  $\rightarrow$  相同下标

{ 结链：相同下标放入同一个链表中 (集合)  
开放地址：用数组其他下标存放

## 直接存取

处理问题所在全集  $U = \{0, 1, \dots, u-1\}$ ,  $|U| = u$  不会特别大

字型 ASCII  $\{0, 1, \dots, 127\}$

数组  $H$  每个位置  $\rightarrow$  格 (slot) 代表  $U$  中的一个键

结构体  $\begin{cases} \text{value} & \text{值} \\ \text{empty} & \text{空否} \end{cases} \Rightarrow H$

查找  $k$ :  $\text{return } !H[k].\text{empty}$

放入  $(k, v)$   $\begin{cases} H[k].\text{value} = v; \\ H[k].\text{empty} = \text{false}; \end{cases}$

删除  $(k, v)$   $H[k].\text{empty} = \text{true};$

$O(1)$  时间完成

0		✓
1		✓
2	Jack	X
3		✓
4	Kate	X

直接存取

位向量形式更紧凑

..... 1个字节

8个元素

empty



# 散列表

宇宙量  $U$  很大, 元素个数不多

一切数字记录 ( $B$  个字节) 都可以表示为  $U = \{0, 1, \dots, 2^B - 1\}$ ,  $|U|$  会非常大

键集合  $K \subset U$   $|K|/|U|$  不太小 存储空间  $\Theta(|K|)$   $c \times |K|$  空间

保证平均情况  $k$  的操作  $O(1)$  时间.

散列函数  $M = \{0, 1, \dots, m-1\}$   $|M| \ll |U|$

$h: U \rightarrow M$   $key \Rightarrow h(key)$  散列值

数组长度  $|U| \rightarrow |M|$   $(k, v)$  被散列到  $h(k)$  中

碰撞:  $k \neq k'$  有可能  $h(k) = h(k')$  缺点

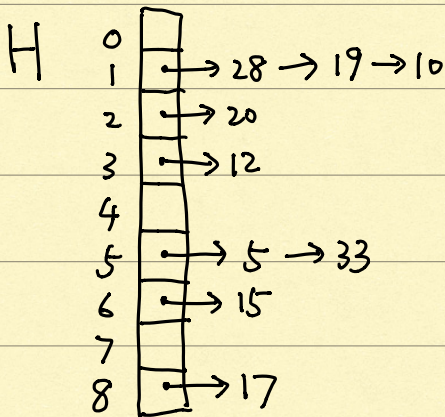
散列要散开 尽可能随机化 但不可避免碰撞 (抽屉原理)

## 结链法

相同散列值的元素放入链表 (也可放入白黑、红黑树之中)

KV 型结构体  $(k, v)$  散列  $vector<vector<KV>> H;$

$H[i] \Rightarrow vector<KV>$   $vector<list<KV>> H;$



$$h(k) = k \bmod 9$$

5, 28, 19, 15, 20, 33, 12, 17, 10

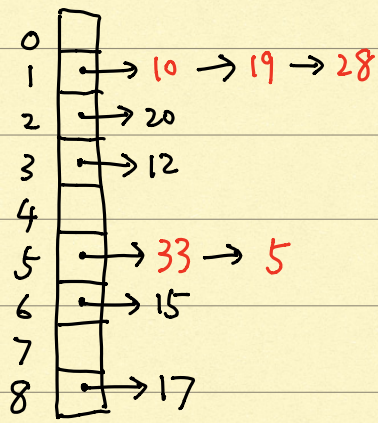
查找:  $H[h(k)]$  中是否有  $k$   $O(n)$  最坏

放入: 若  $H[h(k)]$  中没有  $k$ , 则放入

→ 末尾: 白黑, 双向链表  $O(1)$

→ 前位: 单链  $O(1)$





← 栈形态 (单链表)  
理论分析

删除: 链中删除

双向删除 (无序, 补填充  $\Rightarrow$  性能变差)

缓冲区 (8个元素)

$O(1)$

