

哈希

哈希函数

- 哈希函数也叫散列函数，它对不同的输出值得到一个固定长度的消息摘要。理想的哈希函数对于不同的输入应该产生不同的结构，同时散列结果应当具有同一性（输出值尽量均匀）和雪崩效应（微小的输入值变化使得输出值发生巨大的变化）。

冲突解决

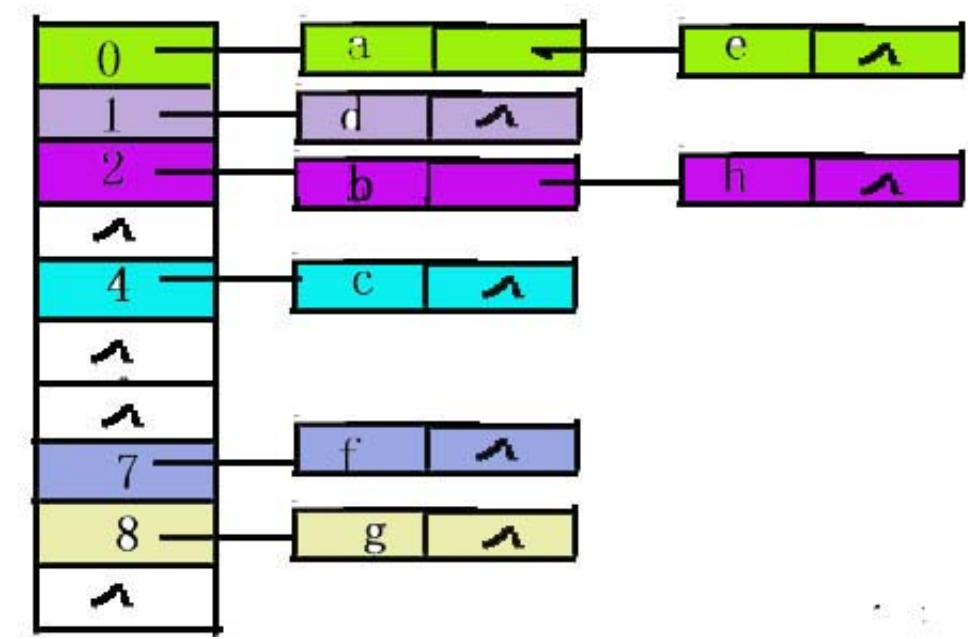
- 现实中的哈希函数不是完美的，当两个不同的输入值对应一个输出值时，就会产生“碰撞”，这个时候便需要解决冲突。

常见的冲突解决方法有开放定址法，链地址法，建立公共溢出区等。实际的哈希表实现中，使用最多的是链地址法

链地址法

- 链地址法的基本思想是，为每个 Hash 值建立一个单链表，当发生冲突时，将记录插入到链表中。

例 2 设有 8 个元素 { a,b,c,d,e,f,g,h }，采用某种哈希函数得到的地址分别为：{ 0 ， 2 ， 4 ， 1 ， 0 ， 8 ， 7 ， 2 }，当哈希表长度为 10 时，采用链地址法解决冲突的哈希表如下图所示：



开放地址法

- 1 开放地址法

这个方法的基本思想是：当发生地址冲突时，按照某种方法继续探测哈希表中的其他存储单元，直到找到空位置为止。这个过程可用下式描述：

$H_i(\text{key}) = (H(\text{key}) + d_i) \bmod m \quad (i = 1, 2, \dots, k \quad (k \leq m - 1))$ 其中： $H(\text{key})$ 为关键字 key 的直接哈希地址， m 为哈希表的长度， d_i 为每次再探测时的地址增量。

采用这种方法时，首先计算出元素的直接哈希地址 $H(\text{key})$ ，如果该存储单元已被其他元素占用，则继续查看地址为 $H(\text{key}) + d_2$ 的存储单元，如此重复直至找到某个存储单元为空时，将关键字为 key 的数据元素存放至该单元。增量 d 可以有不同的取法，并根据其取法有不同的称呼：

(1) $d_i = 1, 2, 3, \dots$ 线性探测再散列；

(2) $d_i = 1^2, -1^2, 2^2, -2^2, k^2, -k^2, \dots$ 二次探测再散列；

(3) $d_i =$ 伪随机序列 伪随机再散列