什么是哈希表

关键字为key的元素放到数组中下标是h(key)%length的单元内.

elem.key 🡪 Arry[h(elem.key)%arry.length]

**把Key通过一个固定的算法函数,即所谓的哈希函数转换成一个整型数字，然后就将该数字对数组长度进行取余，取余结果就当作数组的下标，将value存储在以该数字为下标的数组空间里,**

**当使用哈希表进行查询的时候，就是再次使用哈希函数将key转换为对应的数组下标，并定位到该空间获取value，如此一来，就可以充分利用到数组O(1)的定位性能进行数据定位。**

散列冲突

两个元素关键字相同,映射到数组的同一单元.

解决冲突的方法

1. 开放地址法
2. 链接法
3. 公共溢出区

影响哈希表性能的因素

装填因子

装填因子α = 哈希表元素总数/数组长度 = 平均每个数组单元后挂元素个数.

α越大,产生冲突的可能性越大,性能越差.

优化:初建哈希表时,设置数组容量大些. 哈希表使用中, α增大到一定程度,把哈希表中的数据转移到更大的哈希表中.

散列函数

一个优秀的散列函数满足:计算结果随机,元素均匀分布在数组. 一个坏的散列函数,最差情况,所有元素映射到同一个单元,时间复杂度是O(n).

证明哈希表查找,插入,删除的时间复杂度是O(1)

1,链接法,采用双向链表

2,初创哈希表使用的数组长度与哈希表元素个数成正比

证明:

设哈希表,n个元素,数组长度len.查找长度是1+n/len.因为len=k\*n,所以查找长度是1+1/k.

单次查找长度是常数,不随输入规模n变化,所以时间复杂度是O(1).

哈希表的缺点

哈希表基于数组,一旦创建,难于扩展,填装因子增大到一定值,查找性能下降严重,所以创建哈希表时,应该对将来的数据规模心中有数.

几乎不能有序遍历

哈希表的应用

补充说明