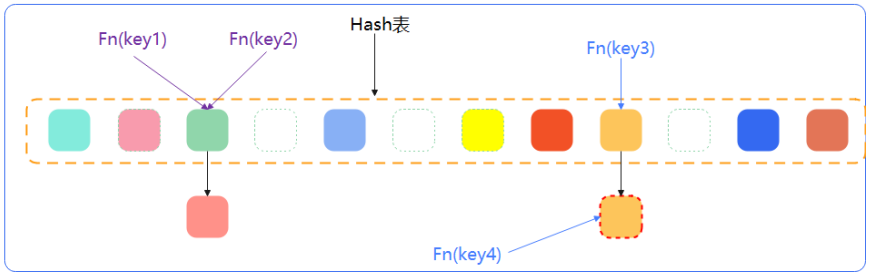
【第04话：当面试官问起什么是Hash碰撞，如何解决Hash碰撞时这样说比较好】

什么是Hash碰撞，如何解决Hash碰撞(月薪1万的回答示范)

1、Hash表就是数组
2、Key经Hash函数计算放入Hash表
3、不同值经Hash函数运算后得到相同结果就是Hash碰撞
4、可通过添加链表解决碰撞



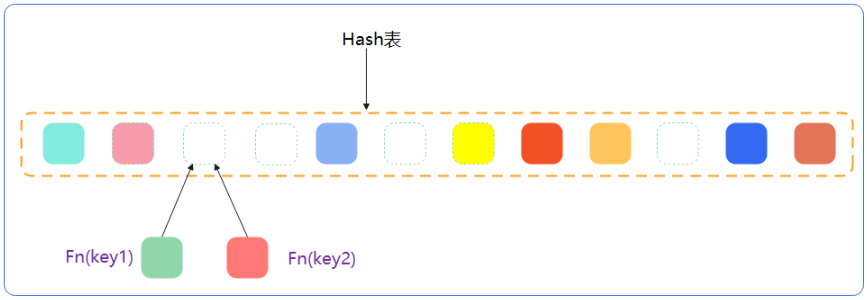
Hash表就是数组变形。Hash碰撞就是向Hash表中存储内容时，不同的内容经过Hash函数计算后，定位到Hash表中相同位置。

解决Hash碰撞可以在数Hash表中碰撞位置放置一个链表。碰撞后把值放置在链表中。Java中HashMap就是对Hash表的实现，底层就是使用了数组+链表+红黑树的方式。

以上什么是Hash碰撞，如何解决Hash碰撞月薪1万的回答示范。

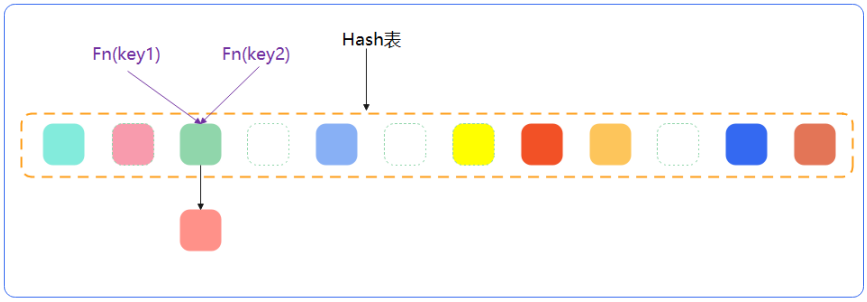
什么是Hash碰撞，如何解决Hash碰撞(月薪1.5万的回答示范)

1、Hash表就是数组的变形
2、Hash函数负责计算内容存放位置
3、Hash碰撞：不同值运算后结果相同
4、解决Hash碰撞常见：
    链式地址法
    开放地址法
    再次Hash法
    建立公共溢出区



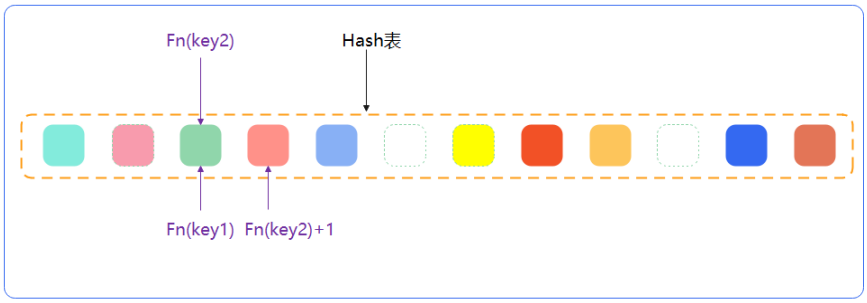
Hash碰撞

1、链式地址法，又叫拉链法
2、碰撞后形成链表



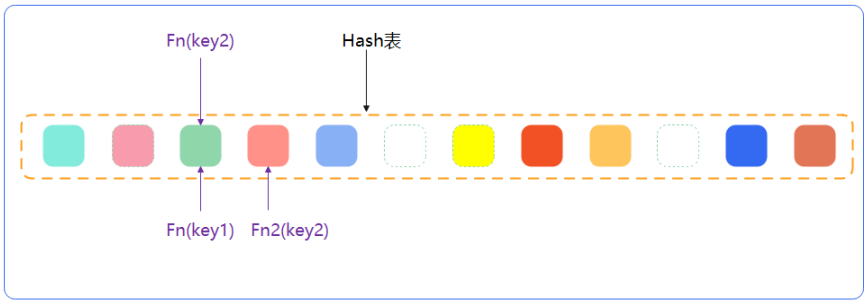
链式地址法

1、开放地址法只有Hash表
2、碰撞后按照固定规则再次计算



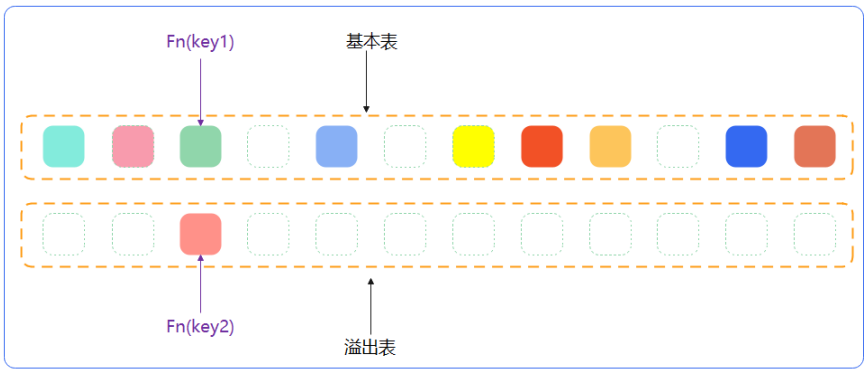
开放地址法

1、再次Hash法只有Hash表
2、需要提前准备多个Hash函数。Fn 、Fn2、Fn3
3、碰撞后使用另一个函数



再次Hash法

1、除了基本表，还需要有溢出表
2、碰撞后放到溢出区



建立公共溢出区

Hash表就是数组变形。通过Hash函数对内容进行运算，得到存储在数组的脚标。虽然Hash函数有很多种。正常情况下，希望不同内容经过同一个Hash函数运算后的结果不同。但是有一种概率事件，就是不同的值经过同一个Hash函数得到了相同结果，这就是Hash碰撞，又叫Hash冲突。

解决Hash碰撞的方式有很多种，但是常见的就四种：链式地址法、开放地址法、再次Hash法、建立公共溢出区。

其中链式地址法，又叫拉链法。就是Hash表中每个桶发生碰撞后，碰撞值形成一个链表。Java中HashMap就是使用的链式地址法，只是在1.8版本以后对链式地址法做了一个升级，链表长度大于8后会转换为红黑树。

开放地址法是当发生碰撞了，按照固定规则再次对内容进行计算，把计算结果放在一个没有内容的桶中，如果桶中依然有内容，再次按照这个规则计算，直到找到没有数据的桶中。

再次Hash法提前准备好多个Hash函数。当使用第一个Hash函数计算后发现冲突，就使用第二个Hash函数，直到不再冲突。

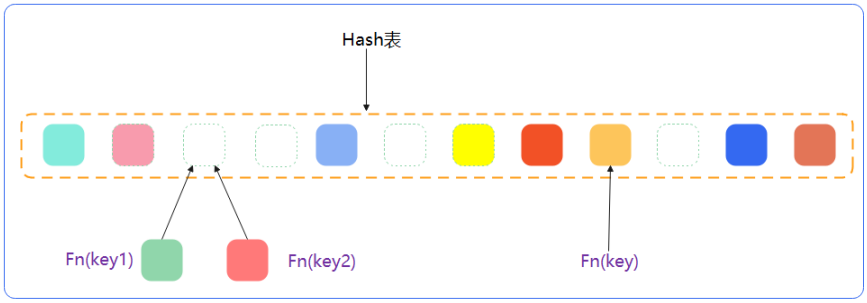
建立公共溢出区把hash表分为基本表和溢出表。当基本表冲突后，把内容放在溢出表中。

以上什么是Hash碰撞，如何解决Hash碰撞月薪1.5万的回答示范。

什么是Hash碰撞，如何解决Hash碰撞(月薪2万+的回答示范)

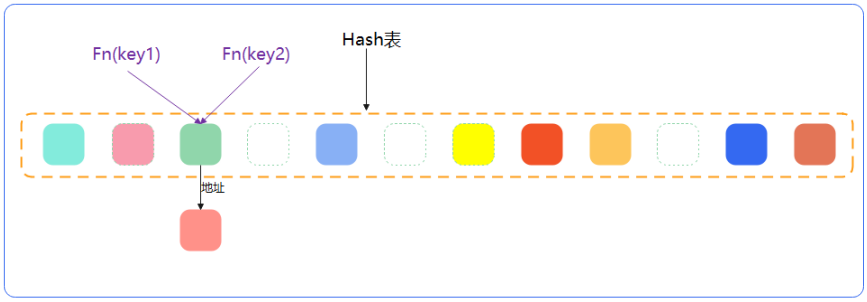
1、无序数组O(n)
2、有序数组O(logn)
3、Hash表最快O(1)

1、直接地址法：f(key)=a*key+b
2、除留余数法：f(key)=key%p
3、随机数法：f(key)=random(key)
4、解决Hash碰撞常见：
链式地址法
开放地址法
再次Hash法
建立公共溢出区



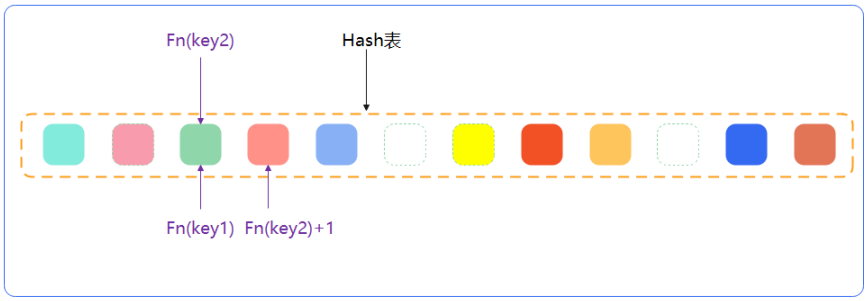
Hash表

1、链式地址法，又叫拉链法
2、碰撞后形成链表
3、使用Hash表长度不固定场景
4、链表长度影响查询新增



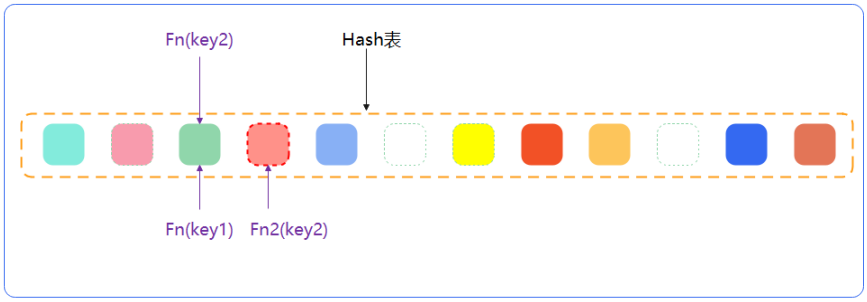
链式地址法

1、开放地址法只有Hash表
2、公式：Fi=(F(Key)+di) %m i=1,2,...,k(k<= m-1)
3、线性探测再散列:di=1,2...m-1
4、二次探测再散列:di=1*1,-1*1..k*k,-k*k(k<m/2)
5、双散列法:di=F2(key)
6、随机探测再散列:di=random()
7、删除为标记删除



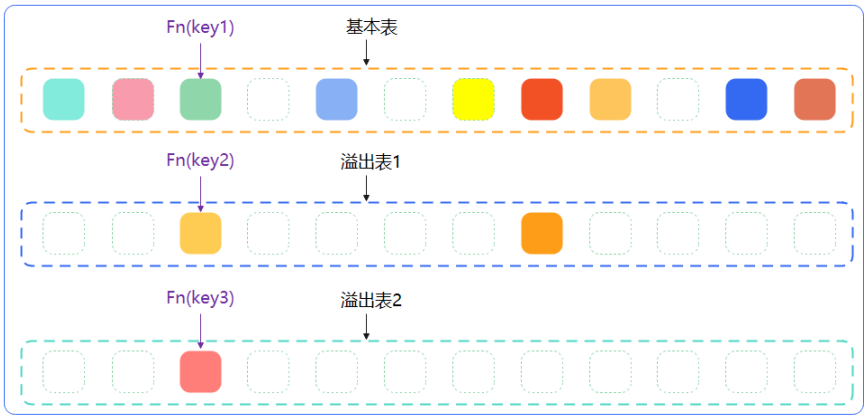
开放地址法

1、再次Hash法只有Hash表
2、需要提前准备多个Hash函数。Fn 、Fn2、Fn3
3、碰撞后使用另一个函数
4、极限情况可能出现函数不够



再次Hash法

1、除了基本表，还需要有溢出表
2、碰撞后放到溢出区
3、随着溢出表增加，浪费空间增大



建立公共溢出区

在无序数组中查找元素时间复杂度O(n)。即使是有序数组，使用折半查找，时间复杂度也才是O(logn)。

Hash表就是数组的变形。最快时间复杂度为O(1)。

Hash表都是通过对Key做Hash函数运算，定位到Hash表，即数组的Bucket中。常见的Hash算法有直接地址法，f(key)=a*key+b，其中a最好是质数,能做到尽量散列，减少碰撞，Java中String的Hashcode就使用质数31实现的;除留余数法f(key)=key%p。随机数法，f(key)=random(key);取随机数。除了这个三个还有很多种Hash函数。

但是无论使用哪种Hash算法，都可能出现不同Key经过同一种Hash函数运算后得到相同的结果，这就是Hash碰撞，又叫Hash冲突。

解决Hash碰撞的方式有很多种，但是常见的就四种：链式地址法、开放地址法、再次Hash法、建立公共溢出区。

其中链式地址法，又叫拉链法。就是Hash表中每个桶发生碰撞后，碰撞值形成一个链表。当发生Hash冲突时会把冲突的内容按照顺序放入到链表中。这种方式特别适合Hash表长度不确定的场景。且不要求Hash表长度必须比较大，因为碰撞后可添加到链表中。但是链表中每个节点空间中都需要存储下个节点地址，如果Hash表中元素比较多且Hash表长度较小时，链表长度可能较长，会有大量存储下个节点地址的空间浪费，降低了查询速度。所以拉链法中都会判断链表长度，当链表长度过长时，会扩大Hash表长度，或者把链表转换为红黑树以提升查询性能。

开放地址法是当发生碰撞了，按照固定规则再次对内容进行计算，直到一个没有存储数据的桶中。有一个公式：Di=(D1(Key)+di） %m　i=1,2,...,k(k<= m-1)。公式中m是Hash表长度。di是增量序列。根据增量序列取值不同，分成多种情况。如果di=1,2,3...m-1称为线性探测再散列。如果di=`1*1,-1*1`....`k*k,-k*k`(k<m/2)称为二次探测再散列。如果di=F2(key)称为双散列法。如果di=random()称为随机探测再散列。开放地址法Hash表长度越长，碰撞可能性越低，在散列次数越少。但是更长的散列表会占用更多的空间。而且开放地址法删除时只能做标记删除，而不能置空，否则会导致查询失败。这也导致开发地址法时Hash表值数量只增不减。

再次Hash法需要提前准备好多个Hash函数。当使用第一个Hash函数计算后如果发生冲突，就使用第二个Hash函数，直到不再冲突。这种方式需要提供多个Hash函数，增加了计算的过程，降低了性能。查询时定位到桶后，需要判断桶中Key是否和查询Key相当，不等需要使用第二个Hash函数，依次类推，直到找到这个值。一定程度上降低了查询性能。而且极端情况下，可能出现Hash函数不够用，一直冲突的问题。

建立公共溢出区把hash表分为基本表和溢出表。当基本表冲突后，把内容放在溢出表中。这种方式导致Hash表所占空间以倍数增加。可能需要1个溢出表或N个溢出表。第N个溢出表中可能只使用了一个桶。基本表同一个桶碰撞次数越多，空间浪费越大。

以上就是什么是Hash碰撞，如何解决Hash碰撞，月薪2万+的回答示范。