散列表（中）

面试题目：如何设计一个工业级的散列函数?

思路：

何为一个工业级的散列表？工业级的散列表应该具有哪些特性？结合学过的知识，我觉的应该有这样的要求：

1.支持快速的查询、插入、删除操作；

2.内存占用合理，不能浪费过多空间；

3.性能稳定，在极端情况下，散列表的性能也不会退化到无法接受的情况。

方案：

如何设计这样一个散列表呢？根据前面讲到的知识，我会从3个方面来考虑设计思路：

1.设计一个合适的散列函数；

2.定义装载因子阈值，并且设计动态扩容策略；

3.选择合适的散列冲突解决方法。

知识总结：

一、如何设计散列函数?

1. 要尽可能让散列后的值随机且均匀分布，这样会尽可能减少散列冲突，即便冲突之后，分配到每个槽内的数据也比较均匀。

2. 除此之外，散列函数的设计也不能太复杂，太复杂就会太耗时间，也会影响到散列表的性能。

3. 常见的散列函数设计方法：直接寻址法、平方取中法、折叠法、随机数法等。

设计一个储存字符串的散列函数

hash("nice")=(("n" - "a") * 26*26*26 + ("i" - "a")*26*26 + ("c" - "a")*26+ ("e"-"a")) / 78978

二、如何根据装载因子动态扩容?

1.如何设置装载因子阈值?

①可以通过设置装载因子的阈值来控制是扩容还是缩容，支持动态扩容的散列表，插入数据的时间复杂度使用摊还分析法。

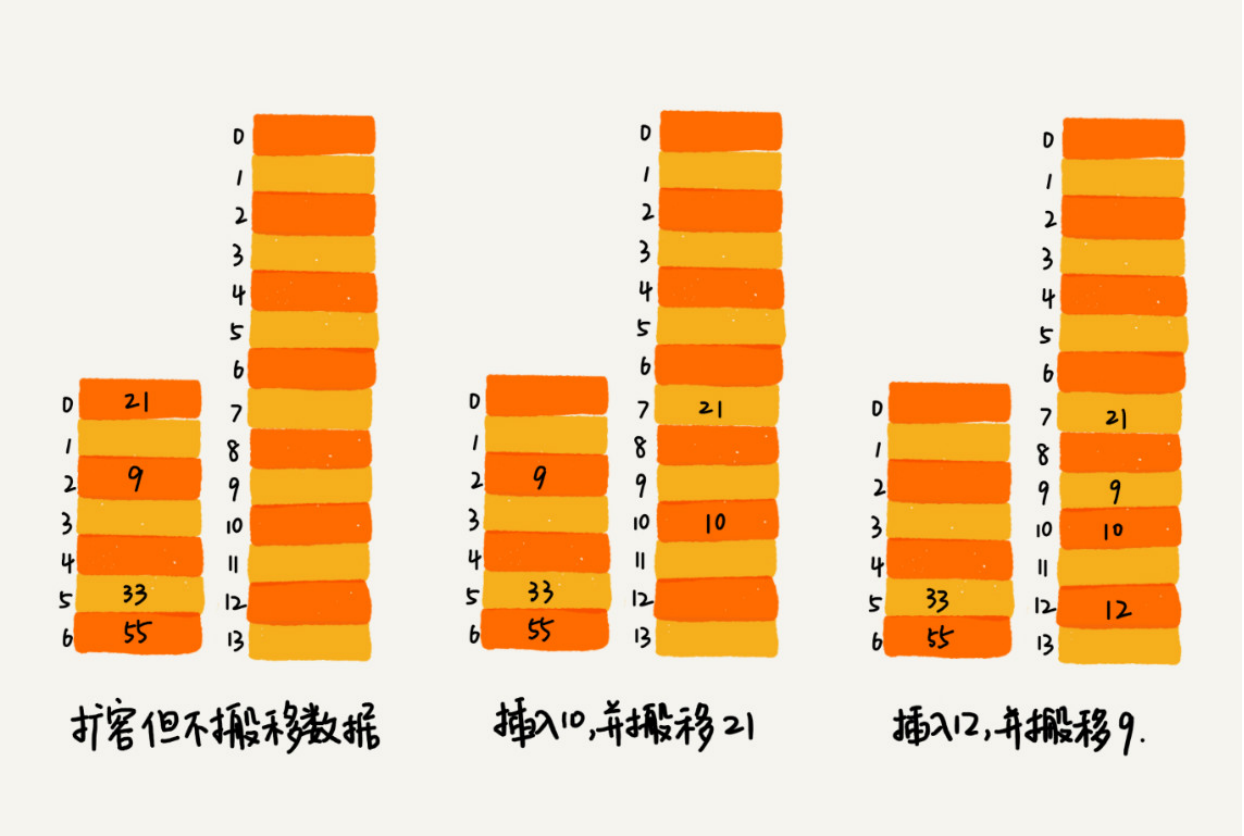
②装载因子的阈值设置需要权衡时间复杂度和空间复杂度。如何权衡？如果内存空间不紧张，对执行效率要求很高，可以降低装载因子的阈值；相反，如果内存空间紧张，对执行效率要求又不高，可以增加装载因子的阈值。

2.如何避免低效扩容？分批扩容

①分批扩容的插入操作：当有新数据要插入时，我们将数据插入新的散列表，并且从老的散列表中拿出一个数据放入新散列表。每次插入都重复上面的过程。这样插入操作就变得很快了。

②分批扩容的查询操作：先查新散列表，再查老散列表。

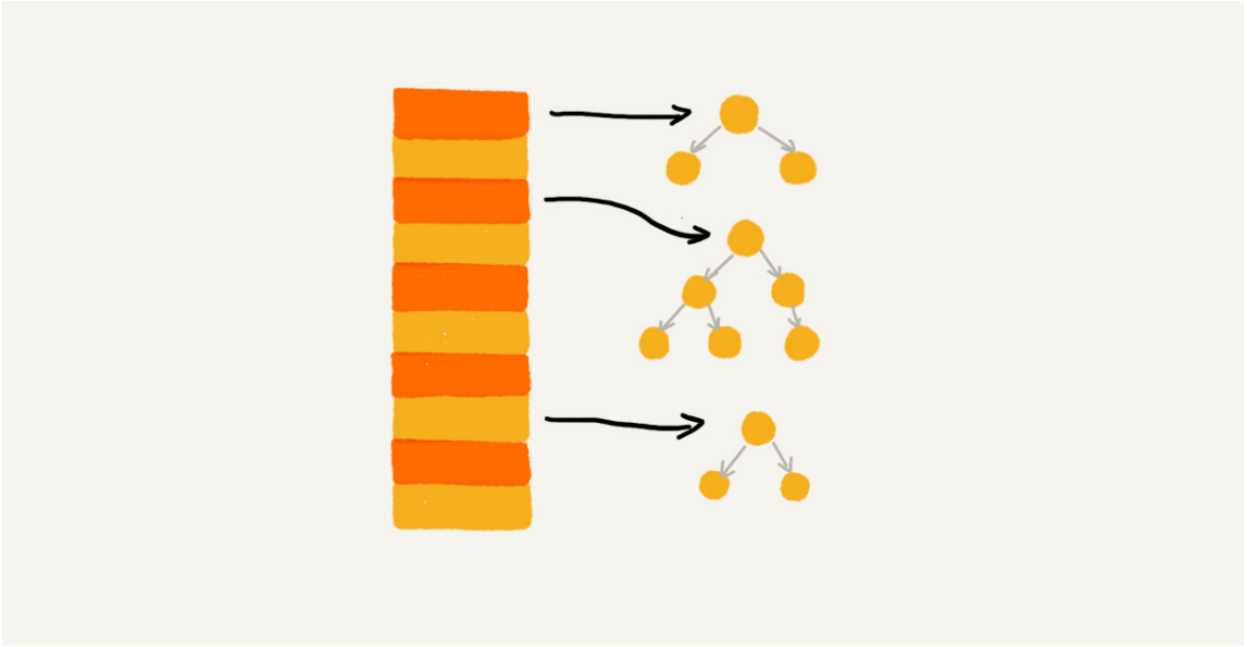
③通过分批扩容的方式，任何情况下，插入一个数据的时间复杂度都是O(1)。



三、如何选择散列冲突解决方法?

①常见的2中方法：开放寻址法和链表法。

②大部分情况下，链表法更加普适。而且，我们还可以通过将链表法中的链表改造成其他动态查找数据结构，比如红黑树、跳表，来避免散列表时间复杂度退化成O(n)，抵御散列冲突攻击。



③但是，对于小规模数据、装载因子不高的散列表，比较适合用开放寻址法。

极客时间文档：https://time.geekbang.org/column/article/64586