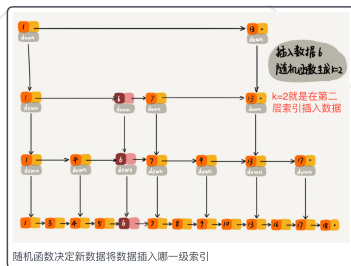


1. 解答:
Redis 中的有序集合支持的核心操作主要有下面这几个:
(1) 插入、删除、查找数据;
(2) 按照区间查找数据 (比如查找值在 [00, 356] 之间的数据);
(3) 迭代输出有序序列
原因:
* (1) 中的操作跳表和红黑树复杂度都一样, 但是 (2) 中的操作跳表效率更高
* 跳表和红黑树相比更容易实现
* 跳表更加灵活, 可以改变索引构建策略来平衡执行效率和内存消耗

4. 练习:
1. redis 的有序集合 (Sorted Set) 为什么使用跳表而不是红黑树?



3. 跳表索引动态更新

12. 跳表

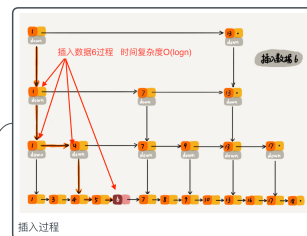
1. 概念与理解

1. 概念
“链表加多级索引的数据结构, 是一种动态数据结构, 支持快速插入、删除、查找操作”是一种以空间换时间的思想, 来提高查询效率, 实现类似二分查找

2. 时间复杂度求解如图

3. 空间复杂度
每 2 个节点建立一个索引:
索引的节点总和就是 $n/2 + n/4 + n/8 + \dots + 8 + 4 + 2 + n/2$, 所以说, 跳表的空间复杂度是 $O(n)$, 也就是说, 如果将包含 n 个结点的单链表构造造成跳表, 我们需要额外再用接近 n 个结点的存储空间
每 3 个节点建立一个索引:
 $n/3 + n/9 + n/27 + \dots + 9 + 3 + 1 = n/2$, 尽管空间复杂度还是 $O(n)$, 空间复杂度还是 $O(n)$, 但是节点数少了一半

2. 跳表动态插入和删除



删除操作:
除了要删除原始链表中的结点, 还要删除索引中的, 因为单链表中的删除操作需要拿到要删除结点的前驱结点, 然后通过指针操作完成删除。所以在查找要删除的结点的时候, 一定要获取前驱结点。当然, 如果我们用的是双向链表, 就不需要考虑这个问题了

