

# 跳表

- 跳表是一种有序表，本质是以空间换时间，在有序链表的基础上维护了多层级的索引结构，基于二分查找方式实现  $O(\log N)$  时间复杂度的增删查改操作
- 红黑树
  - 搜索树的一种，相较于 avl 树，牺牲了一定的平衡性，但使用更少的自旋等操作来维护搜索树的平衡
  - 根节点和叶子节点是黑色
  - 根节点到叶节点的路上，不允许出现两个连续的红色节点
- 红黑树对比跳表
  - 性能方面两者差不多，时间复杂度差不多，空间复杂度的话跳表需要维护冗余的多层索引结构，空间复杂度会稍高，但没有引起量级的变化（比如  $n$  和  $2n$ ）
  - 跳表实现难度远低于红黑树
  - 红黑树基于二叉树，对于范围查询不能很好支持；跳表基于有序单向链表，天然支持范围查询
  - 红黑树由于维护平衡性的手段涉及到节点旋转和染色，并发操作时需要加全局锁；跳表则可以基于更细粒度的锁
- 跳表的查询过程
  - 每层链表都是单向有序链表
  - 第一层链表拥有全部的数据
  - 第  $i+1$  层链表个数为第  $i$  层的  $1/2$ ，且所有节点都在第  $i$  层存在
  - 总高度为  $\log N$
  - 基于以上特性，查询时先从高层查询，在找到 target 或者发现不存在时，选择向右或向下遍历
- mysql 为什么用 b+ 树而不是用跳表
  - 两者本质上都是链表加多级索引组合的结构
  - 但是 b+ 树节点是 page (16k 的数据页)，能存下较多的索引和数据页，磁盘 i/o 次数更少
  - 原生跳表一条数据一个节点一次磁盘 i/o，更适合内存 i/o