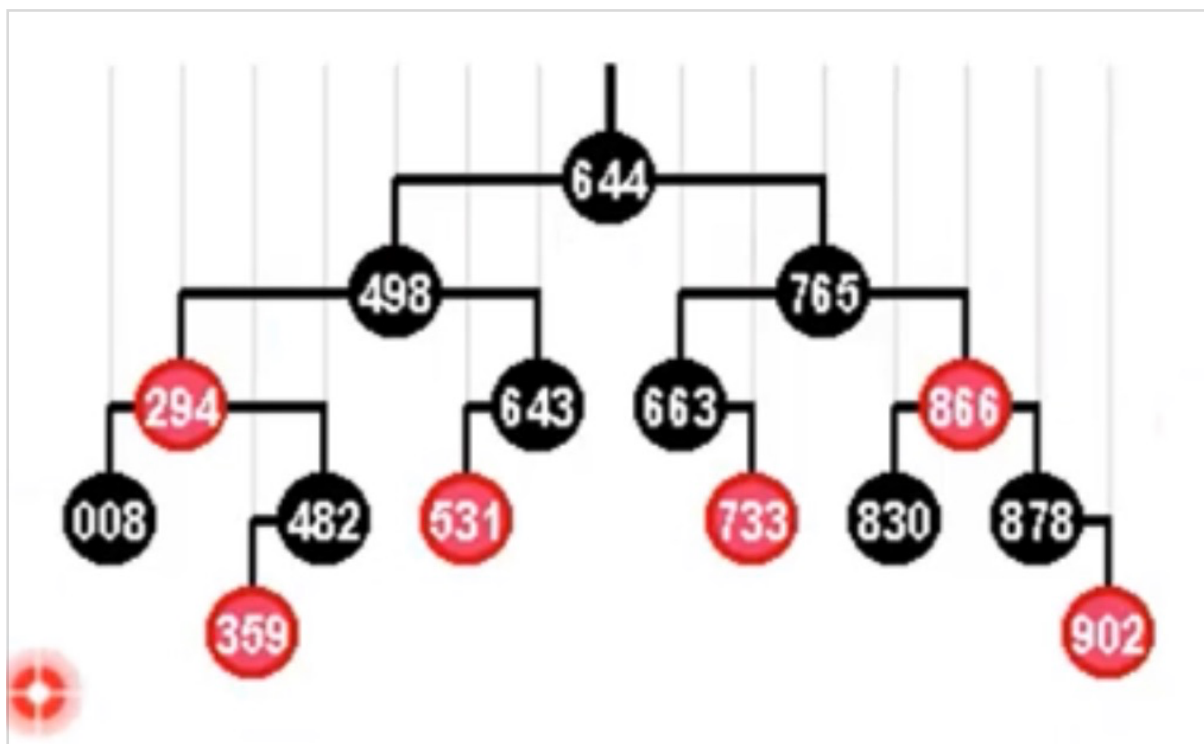


10C1 红黑树：动机

#数据结构邓神

动机



为什么还需要红黑树？

AVL就可以达到 $\log(n)$ 的要求

红黑树的意义是什么呢？

持久性

一致性结构

持久

❖ **Persistent structure**：支持对**历史**版本的访问

`T.search(ver, key); T.insert(ver, key); T.remove(ver, key)`

//ephemeral

❖ 蛮力实现：每个版本独立保存；各版本入口自成**一个搜索结构**

ver.0 ver.1 ver.2 ver.3 ver.4

1 8 1 2 3 7 8 1 2 3 6 7 8 4 5 6 7 8 4 5

❖ 单次操作 $O(\log h + \log n)$ ，累计 $O(h \cdot n)$ **时间** **空间**

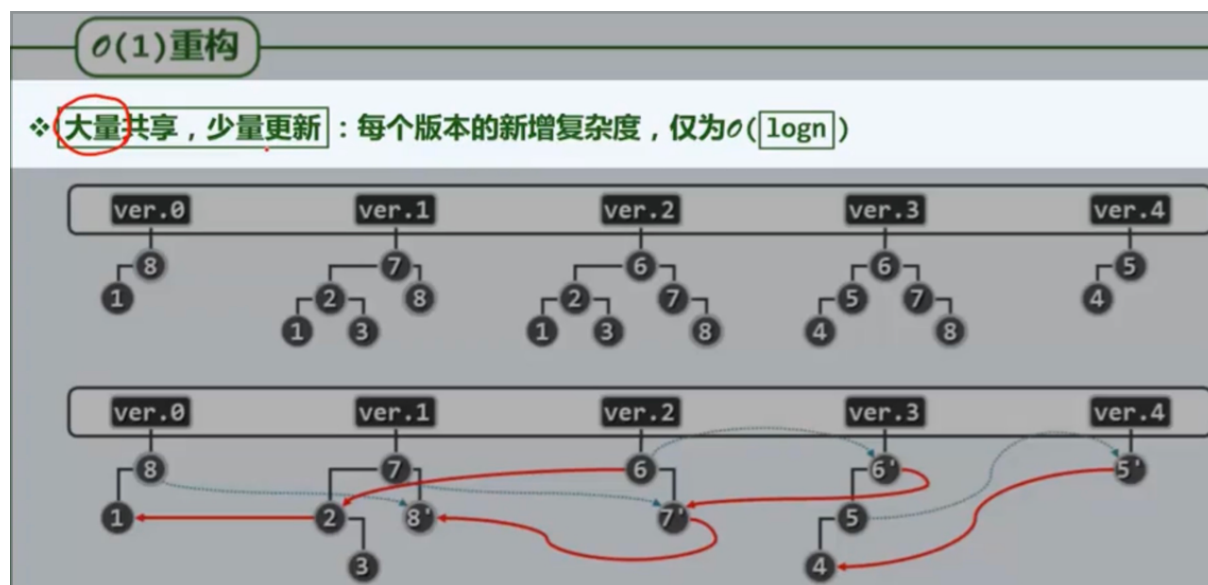
//h = |history|

❖ **挑战**：可否将复杂度控制在 $O(n + h \cdot \log n)$ 内？

可以，我们要利用相邻版本的相关性

关联性

我们是否可以只记录如何变化



能否进一步提高，比如整体 $O(n+h)$ 单版本 $O(1)$?

$O(1)$ 重构



绝大多数的BBST都无法保证这一点，所以我们需要红黑树

拓扑差异主要来自于旋转操作，每一次旋转，都会带来常数级别的差异，也就是说如果要想差异不超过 $O(1)$ 只能进行 $O(1)$ 次旋转，而不是AVL的 $O(\log n)$ 次

**任何一次动态操作
引发的结构变化量
不致超过 $O(1)$**

红黑树

