

## 10A2 双层伸展

#数据结构邓神

### 双层伸展

❖ D. D. Sleator & R. E. Tarjan

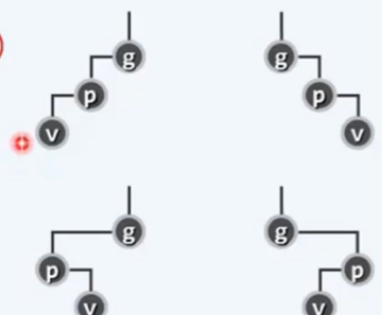

Self-Adjusting Binary Trees

J. ACM, 32:652-686, 1985

❖ 构思的精髓：向上追溯两层，而非一层

❖ 反复考察祖孙三代： $g = \text{parent}(p)$ ,  $p = \text{parent}(v)$ ,  $v$

❖ 根据它们的相对位置，经两次旋转使得  $v$  上升两层，成为（子）树根



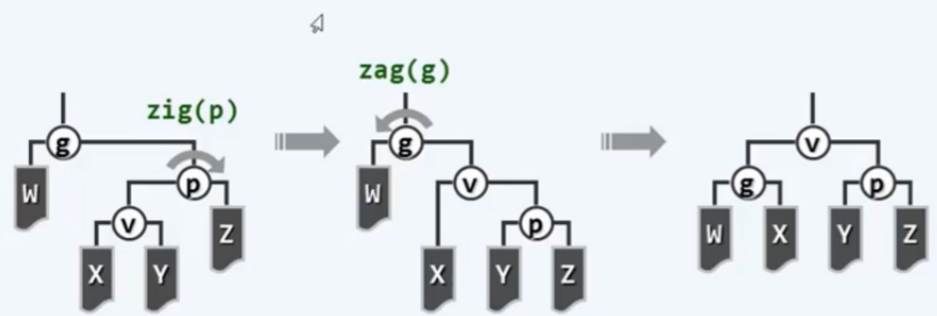
Data Structures (Spring 2014), Tsinghua University

zig-zag | zag-zig

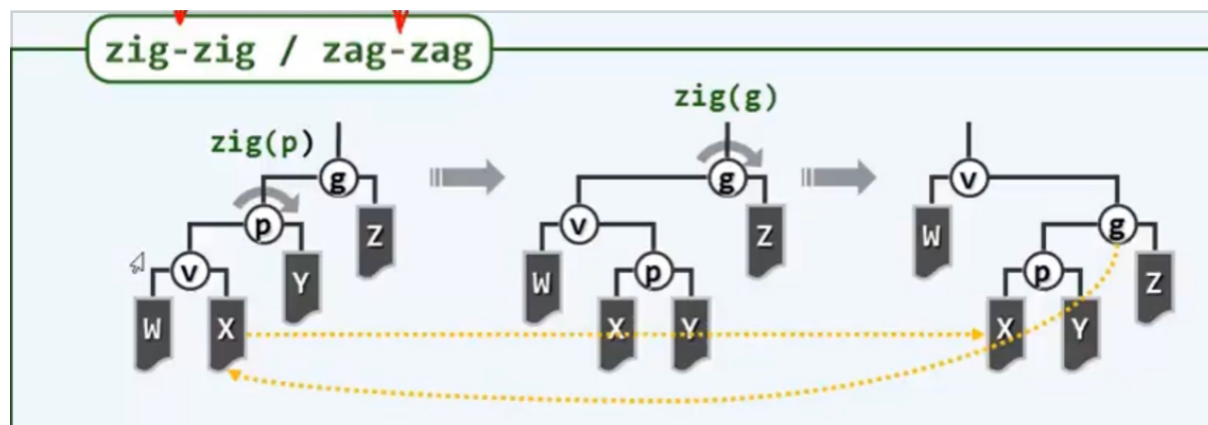
### zig-zag / zag-zig

❖ 与AVL树双旋完全等效！

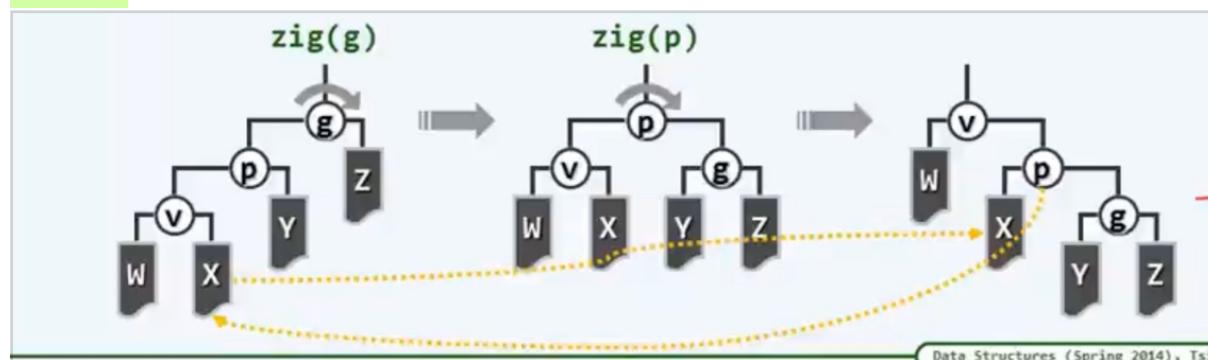
❖ 与逐层伸展别无二致！



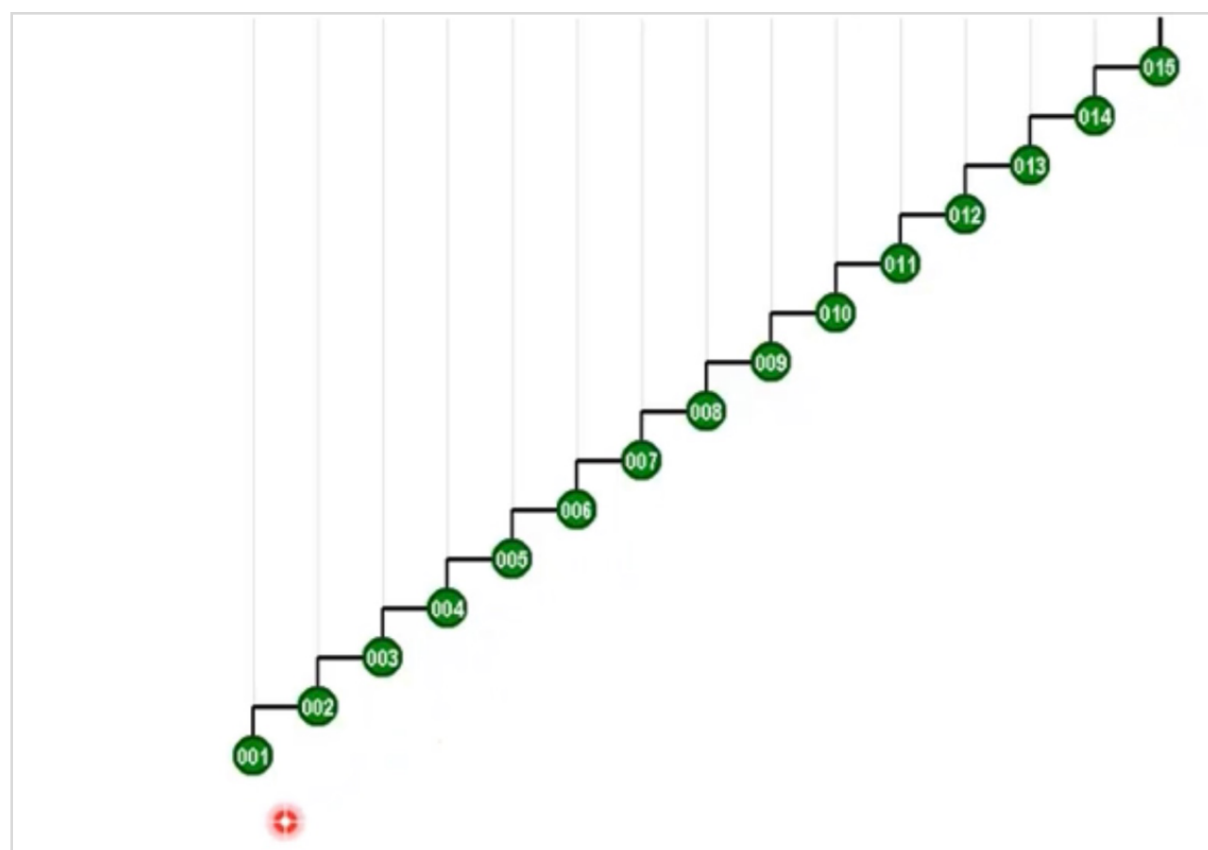
zig-zig | zag-zag

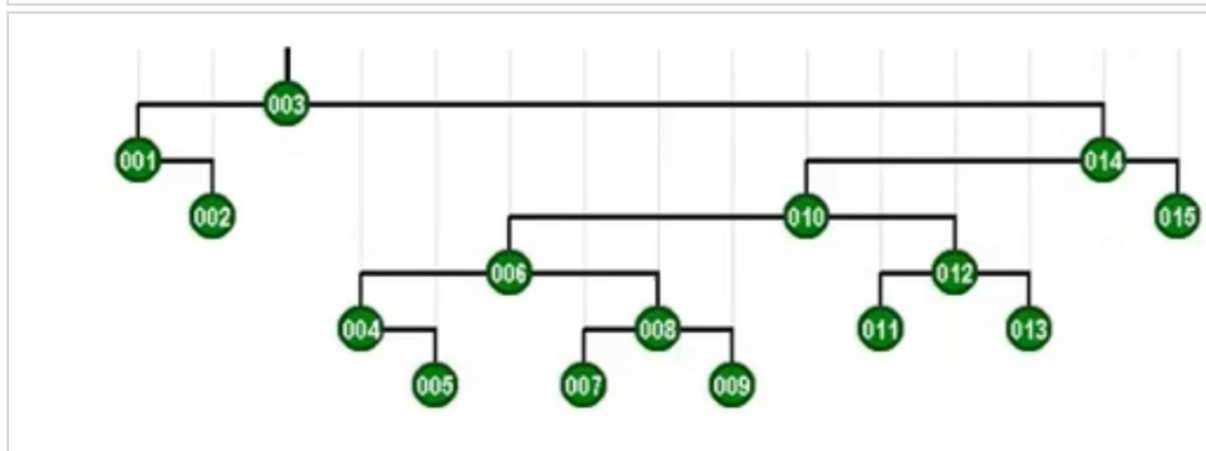
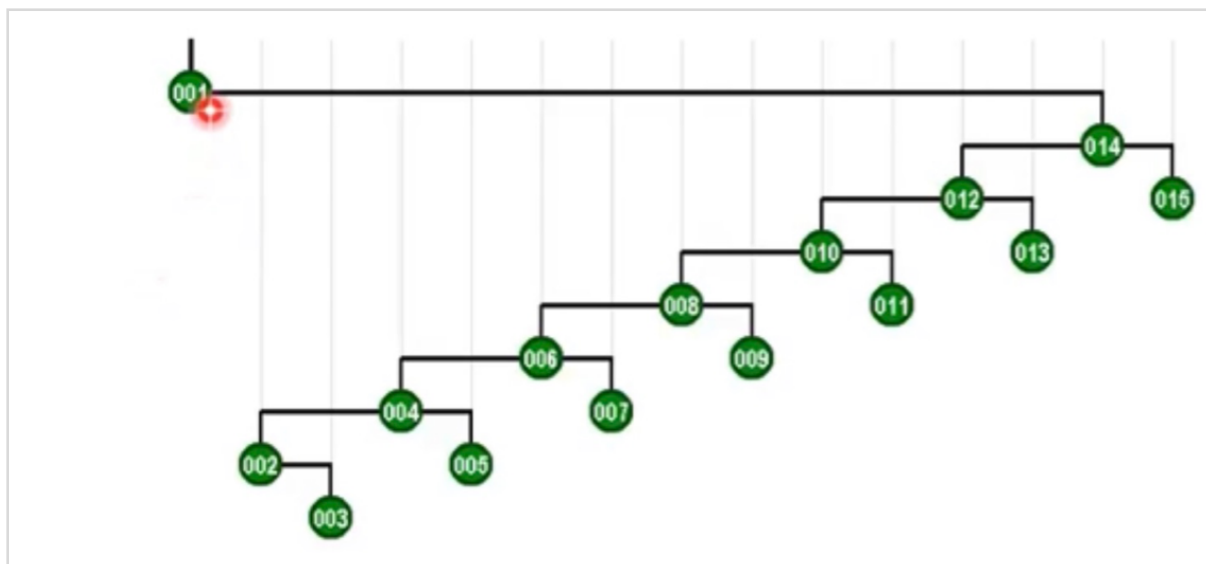


点睛之笔

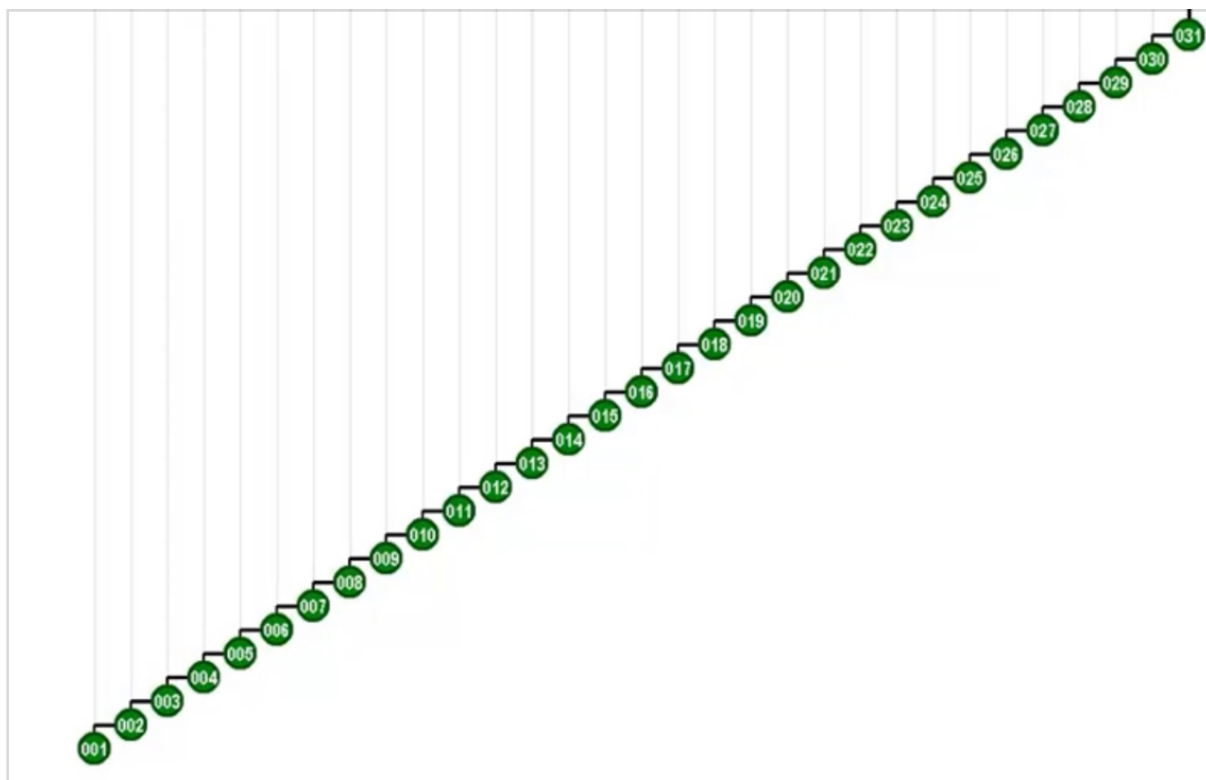


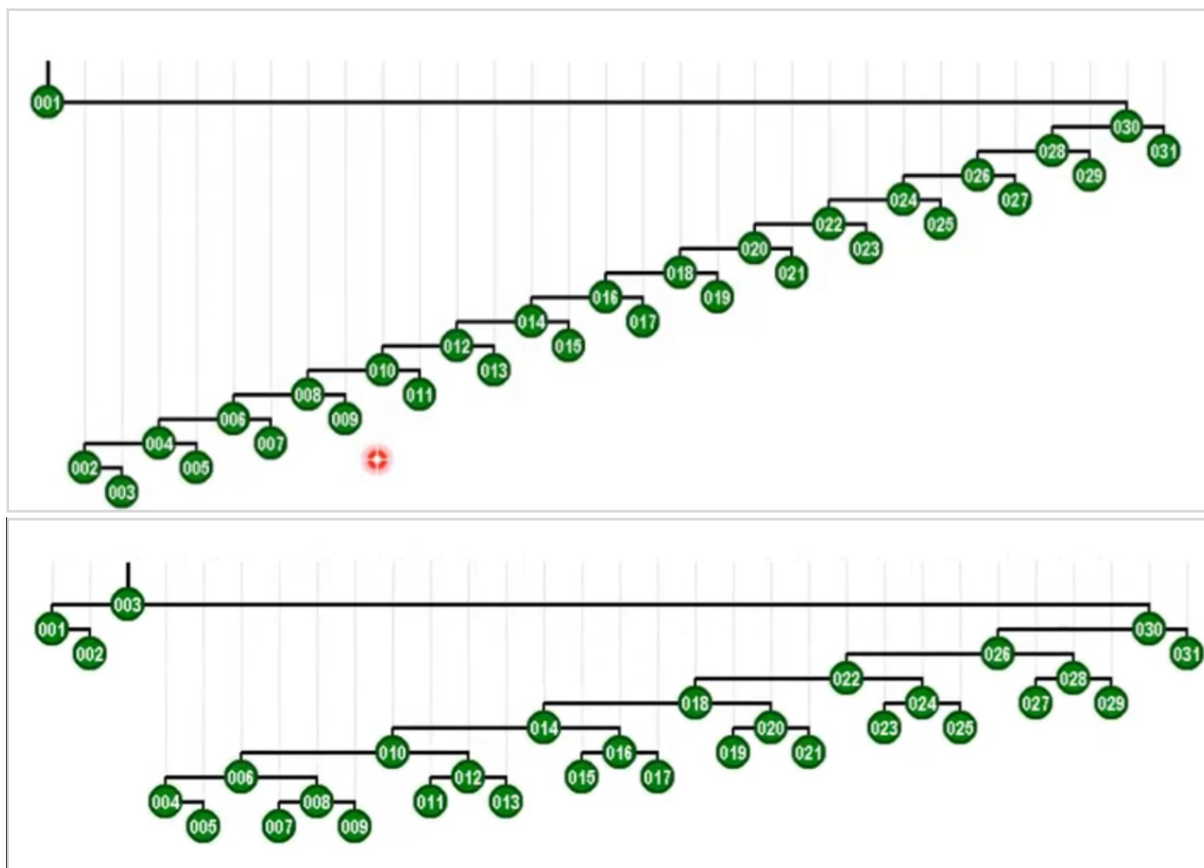
点睛之笔





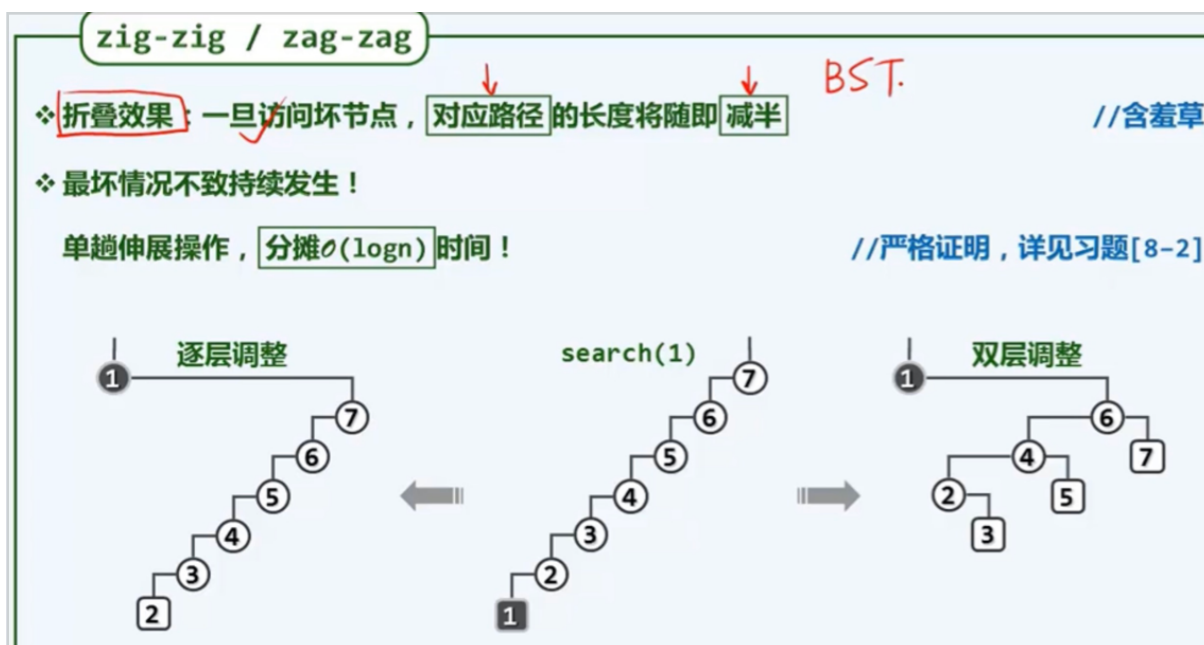
折叠效果





这个数据结构好努力啊

分摊性能



这个数据结构太太太努力啦，不断优化自己的最坏节点😭😭😭

最后一步：

如果  $v$  只有父亲没有祖父会怎么办？

这种情况只会出现一次，很好解决，只需要一次 zig 或者 zag

## zig / zag

❖ 要是v只有父亲，没有祖父呢？

❖ 此时必有  $\text{parent}(v) == \text{root}(T)$ ，且

每轮调整中，这种情况至多（在最后）出现一次

❖ 视具体形态，做单次旋转：zig(r)或zag(r)

