

## 8. 高级搜索树

### (a2) 伸展树：双层伸展

贾政道：“不用全打开，怕叠起来倒费事。”

詹光便与冯紫英一层一层折好收拾。

邓俊辉

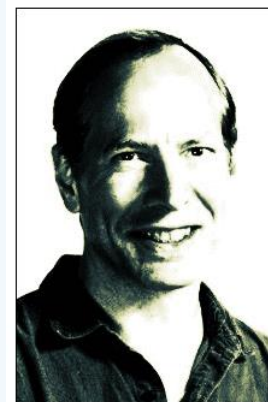
deng@tsinghua.edu.cn

## 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`

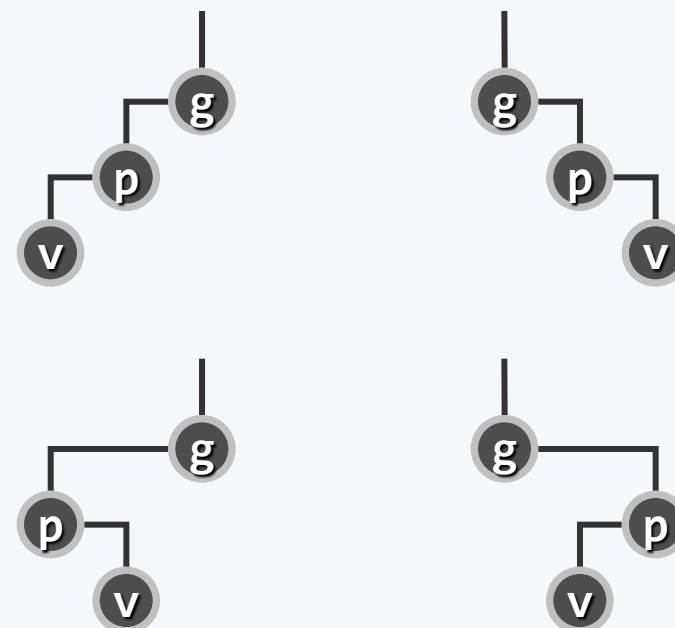
❖ 反复考察`祖孙三代`： $g = \text{parent}(p)$ ,  $p = \text{parent}(v)$ ,  $v$

❖ 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

❖ 如此，性能的确会有改善？

❖ 具体地，应该如何旋转？

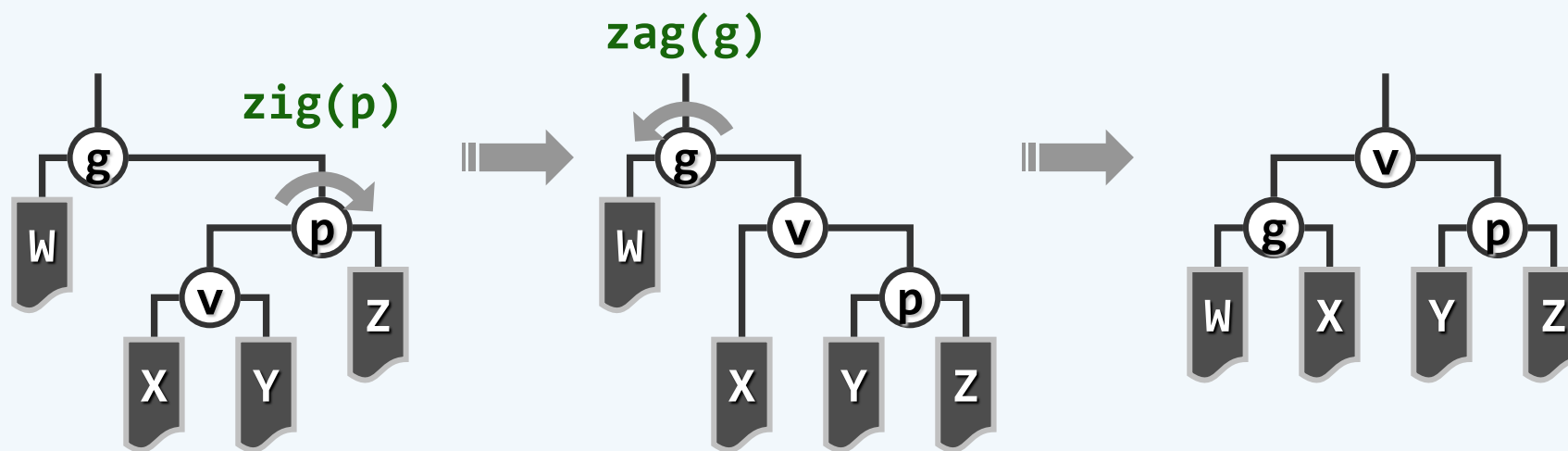


## zig-zag / zag-zig

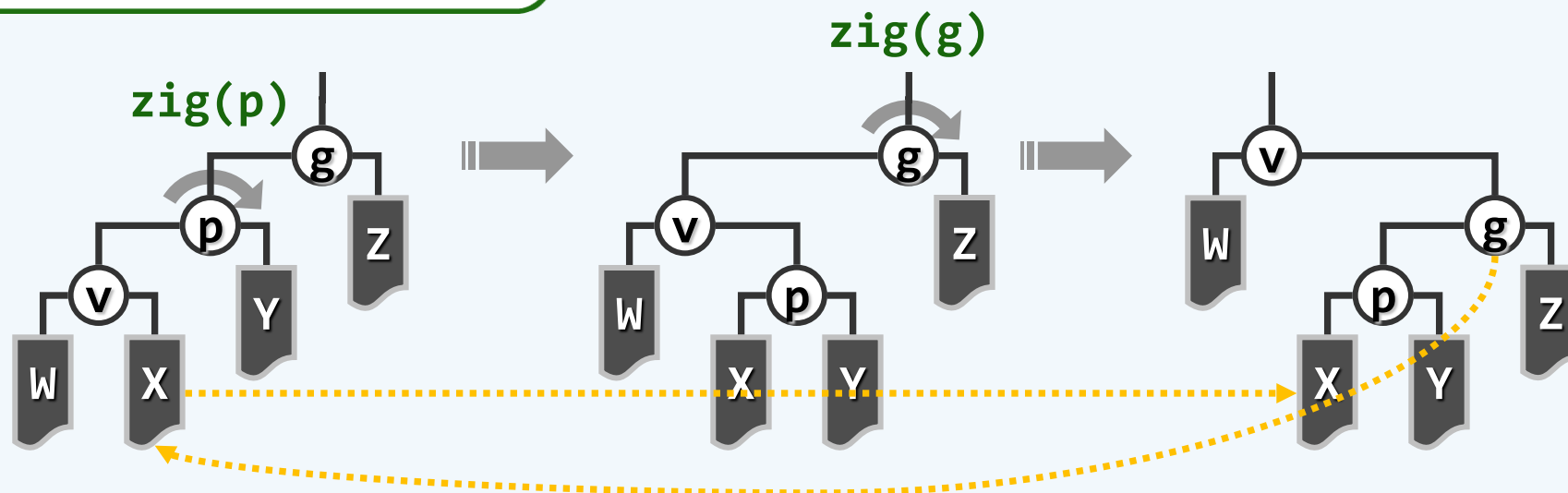
❖ 与 **AVL树双旋** 完全等效！

❖ 与 **逐层伸展** 别无二致！

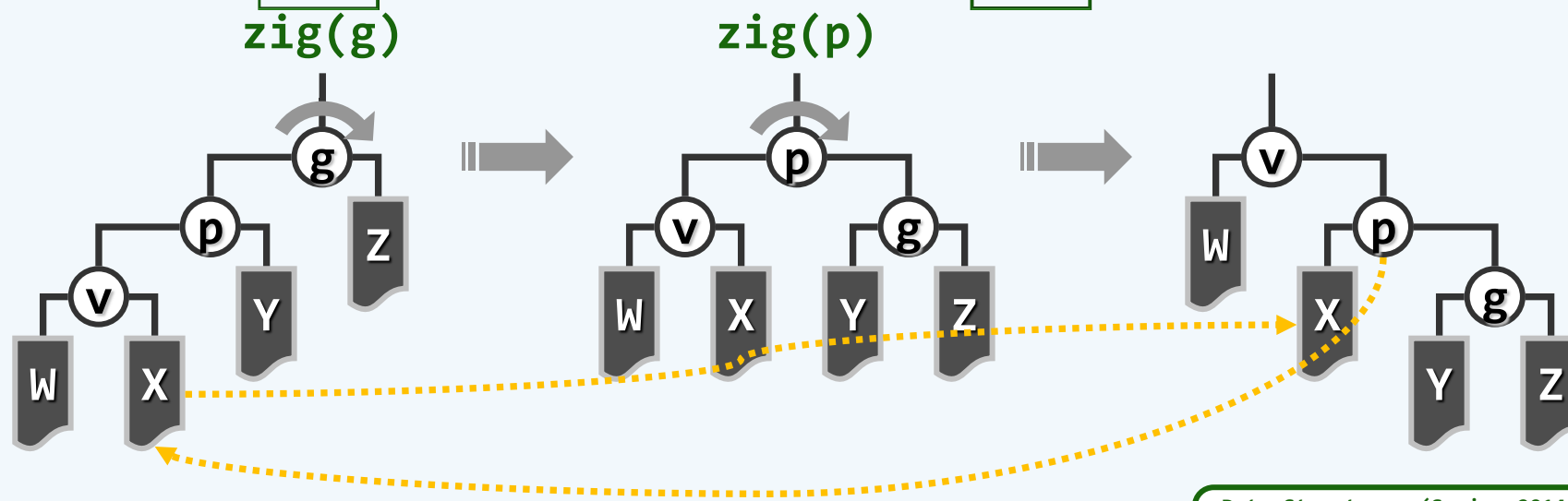
❖ 难道，就这样平淡无奇？



## zig-zig / zag-zag



❖ 颠倒次序之后，**局部**的细微差异，将彻底地改变**整体**...



## zig-zig / zag-zag

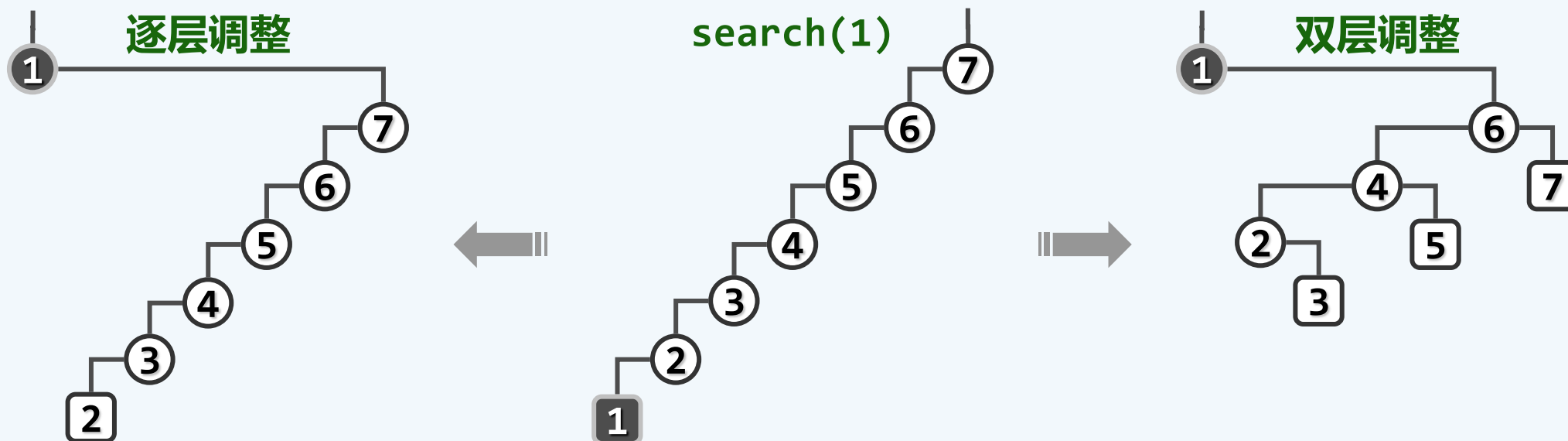
❖ 折叠效果：一旦访问坏节点，**对应路径**的长度将随即**减半**

//含羞草

❖ 最坏情况不致持续发生！

单趟伸展操作，**分摊** $O(\log n)$ 时间！

//严格证明，详见习题[8-2]



## 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： $\text{zig}(r)$ 或 $\text{zag}(r)$

