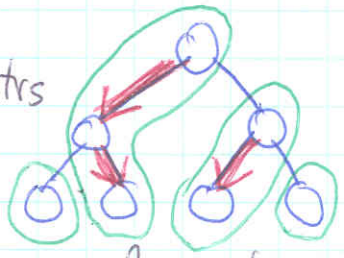
$O(\lg \lg n)$-competitive BST ("Tango trees")

[Demaine, Harmon, Iacono, Pătraşcu - FOCS 2004]

6.897
L 4.4
Feb. 10, 2005

- define preferred child of node y in P to point toward node in y's subtree last accessed, other than y itself
  - either left, right, or none



- preferred path = string of preferred child ptrs
  - pref. paths partition nodes of P

- Idea: store each preferred path in an auxiliary tree
  - "separate", typical balanced BST
  - linked together: leaves of an auxiliary tree point to roots of lower, adjacent auxiliary trees
  - topmost auxiliary tree (containing root of P) is the "starting point" for searches

- search starts in top aux. tree, visits lower & lower aux. trees
  - say k aux. trees in total
  - each aux. tree jump corresponds to traversing a nonpreferred edge connecting two preferred paths
    - ⇒ preferred edge changes
    - ⇒ interleave bound $+1$ (unless there was no preferred edge)
    - ⇒ interleave bound $\geq k-1$ (except for overall cost of n)
  - search cost:
    - each aux. tree has $\leq \lg n$ nodes
    - ⇒ $O(\lg \lg n)$ to search in each aux. tree
    - ⇒ $O(k \lg \lg n)$

⇒ $O(\lg \lg n)$-competitive if updates are cheap.