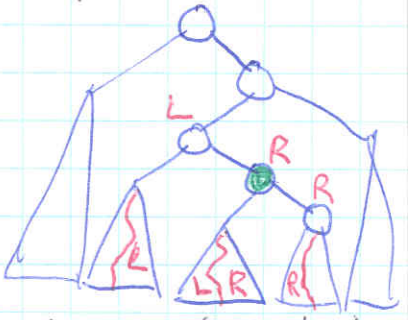
Wilber's first lower bound [SICOMP 1989]

6.897
L 4.3
Feb. 10, 2005

- fix a "lower bound tree" P,
  e.g. a perfect binary tree
- for each node y of P:
  - label each access x_i as
    - L if the key x_i is in y's left subtree,
    - R if the key x_i is in y's right subtree
    - blank if $x_i = y$ or x_i is outside the subtree.
  - interleave(y) = # alternations between L & R
- interleave = $\sum_y$ interleave(y) is a lower bound on all BSTs for X

Proof sketch:

- define transition point of node y in P
  to be the highest node z in BST T such that root-to-z path includes nodes from left & right subtrees of y in P
  - well-defined
  - doesn't change until z is touched
  - different for different nodes y



- for each interleave(y), put marble at transition pt.(y)
- before next interleave(y), must touch this ↑ node
  ⇝ take the marble

⇒ ≤ 1 marble per node in T

⇒ # marbles taken / 6 is a lower bound on # nodes touched

6 interleave − n