CIS 313

week of Oct 22

fifth week of the term
expected behavior

• if list a is chosen randomly from among all n! permutations
• how long does “for i=1 to n T.insert(a_i)” take?
• worst case: O(n^2)
• want to argue: on average O(n lg n)

• main fact: expected search time (1+1/n) in BST built from randomly chosen permutation is 2 \cdot \ln(n + 1) + O(1) \approx 1.38 \log_2 n + O(1)
observations

• this does not bound the height of the tree
• exercise 12.4-2, p 303: describe a binary search tree on n nodes such that the average depth of a node in the tree is $\Theta(lg\ n)$ but the height of the tree is $\omega(lg\ n)$
• stronger result: height of randomly built BST is $\Theta(lg\ n)$

• new goal: maintain BST whose height is $\Theta(lg\ n)$ in the worst case
• self balancing search trees: AVL, red-black, B-trees
balanced tree

• not realistic to expect perfectly balanced tree
• one attempt (not common): *weight-balance*, where the number of nodes in left and right subtrees of any node must be close to each other
• better: *height-balance*, the height of the left and right subtrees must be close
• AVL: differ by one
• red-black: differ by factor of two
• balance maintained by rotations
rotation: single
rotations: double

Composed from two single rotations.
AVL trees

• (not in text)
• named after inventors Adelson-Velskii and Landis
• store at each node the balance factor:
  • $bf(p) = \text{height}(p.lchild) - \text{height}(p.rchild)$
  • requirement: for every node $p$, $bf(p)$ equals -1, 0, or 1
• requires two bits extra storage at each node
AVL insertion

- insert node as with a BST (add it to a null pointer)
- update balance factors along path from new node to root
- the balance factors of some nodes may be in violation: 2 or -2
- find the critical node: the lowest out of balance node
- perform the appropriate rotation

- note: this will affect the balance factors of nodes above it
- total insertion time $O(lg n)$
AVL height is $O(lgn)$

- let $G_k$ be an AVL tree (shape) of height $k$ with the fewest number of nodes
- $G_k$ can be constructed inductively as a node with a $G_{k-1}$ left child and a $G_{k-2}$ right child
- define $g_k$ to be the number of nodes in a $G_k$ tree
  - $g_0 = 1, g_1 = 2, g_k = 1 + g_{k-1} + g_{k-2}$
- sequence: 1, 2, 4, 7, 12, 20
- fact: $g_k = F_{k+3} - 1$