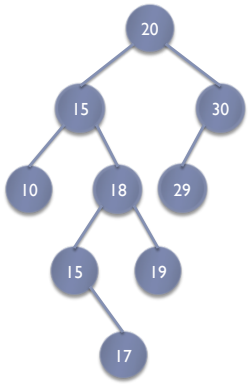
cs140 – algorithms
prof. yi chen

september 26, 2013

Binary search trees

- what is it?
- what operations?
  - search(T,k)
  - minimum(T)
  - maximum(T)
  - successor(T,x)
  - predecessor(T,x)
  - insert(T,x)
  - delete(T,x)

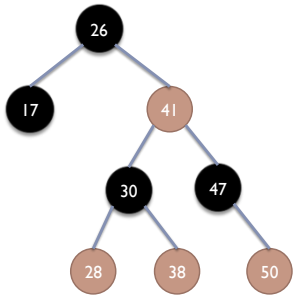


Balanced binary search trees

- Red-black tree
- Splay tree
- 2-3 trees
- AVL trees
- ... etc ...

Red-black trees

1. every node is red or black
2. root is black
3. leaves (null pointers) are black
4. if node is red, both children are black
5. for every node, all paths from node to descendant leaves contain the same number of black nodes.



Red-black trees

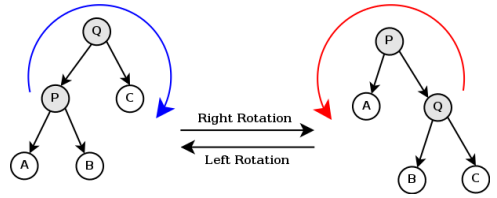
- any node with height h has bh≥h/2
- subtree rooted at any node x contains at least $2^{bh(x)}$-1 nodes
- red-black tree with n internal nodes has height at most 2log(n+1)

1. every node is red or black
2. root is black
3. leaves (null pointers) are black
4. if node is red, both children are black
5. for every node, all paths from node to descendant leaves contain the same number of black nodes.

search(T,k)
minimum(T)
maximum(T)
successor(T,x)
predecessor(T,x)
insert(T,x)
delete(T,x)

Red-black trees

- insertions
- deletions
- key operation: rotations



http://en.wikipedia.org/wiki/Tree_rotation

Mergeable heaps

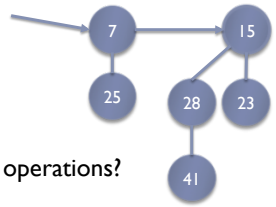
- Operations
  - make-heap()
  - insert(H,x)
  - minimum(H)
  - extractMin(H)
  - union(H1, H2)
  - decreaseKey(H,x,k)
  - delete(H,x)
- Implementation?

Binomial trees

- A binomial tree B_k consists of two binomial trees B_{k-1} that are linked together: the root of one is the leftmost child of the root of the other.
- Properties
  - nodes in a binomial tree?
  - height of B_k?
  - number of nodes at depth i?
  - degree of root?
  - max degree of node in an n-node binomial tree?

Binomial heap

- Collection of binomial trees
- Properties
  - each binomial tree obeys the min-heap property
  - for any nonnegative integer k, there is at most one binomial tree whose root has degree k



- Implementation?
- Running time of mergeable heap operations?