

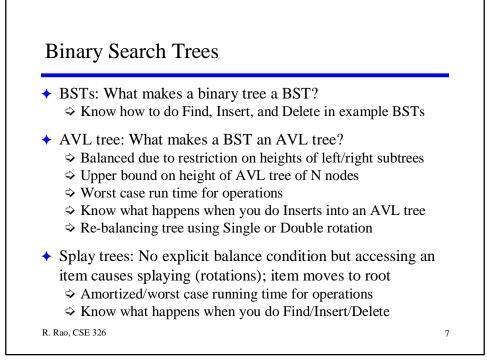
Lists, Stacks, and Queues

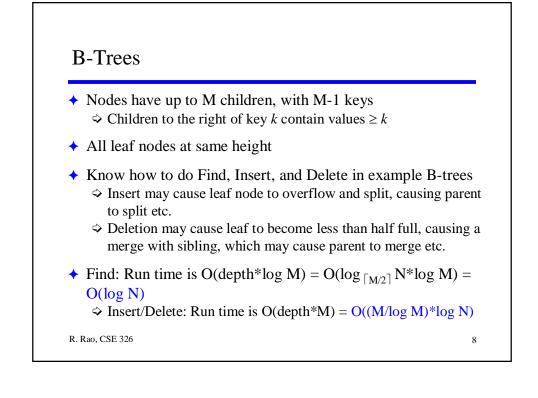
- ✤ Lists: Insert, Find, Delete
 - Singly-linked lists with header node
 - ⇔ Doubly-linked and Circularly-linked
 - ⇔ Run time and space needed for array-based versus pointer-based
- Stacks: Push, Pop
 - Show what push and pop do
 - ⇔ Pointer versus array implementation
 - \Rightarrow Use of stacks in balancing symbols and function calls
- ✤ Queues: Enqueue and Dequeue
 - Array-based implementation using Rear and Front, and modulo arithmetic for wrap-around

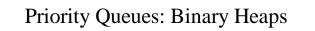
5

R. Rao, CSE 326

+	 Terminology: Root, children, parent, path, height, depth, etc. ⇒ Height of a node is maximum path length to any leaf ⇒ Height of tree is height of root ⇒ Single node tree has height and depth 0
+	Recursive definition of tree ⇒ Null or a root node with (sub)trees as children
+	Preorder, postorder and inorder traversal of a tree ⇒ Implementation using recursion or a stack
+	Minimum and maximum depth of a binary tree







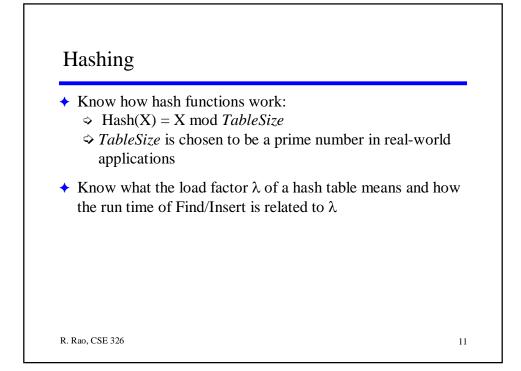
- What is a binary heap?
 - Understand array implementation: parent and children in array
 d-heaps: d children per node
- ✤ Main operations: FindMin, Insert, DeleteMin
 - Show how to Insert/DeleteMin in example binary heaps
 - Solution Insert: Add item to end of array, then *percolate up*
 - DeleteMin: Move item at end of array to top, then *percolate down*
- ♦ Other operations: DecreaseKey, IncreaseKey, Merge
- What is the depth and running time of operations for binary heap of N nodes?
- No need to know details of leftist or skew heaps
 R. Rao, CSE 326

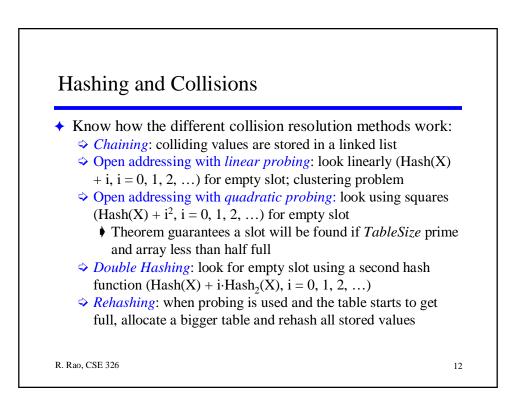


- ◆ Recursive definition of binomial trees
 ⇒ Contains one or more trees B_i, each containing exactly 2ⁱ nodes
- Binomial queue = forest of binomial trees, each obeying heap property
- Main operation: Merge two binomial queues
 Start from i = 0 and attach pairs of B_i to create B_{i+1}
- ✤ Insert item: Merge original BQ with new one-item BQ
- DeleteMin: Delete smallest root node and merge its subtrees with original BQ
- First Child/Next Sibling implementation and run time analysis

R. Rao, CSE 326

9





Next Class: Midterm exam

To Do:

Hash Chapters 1-6 into those good ol' gray cells

Minimize collisions

Practice the practice midterm

R. Rao, CSE 326

13