Intro to Data Structures

Lecture #20 – Binary Search Trees
(implementation)
November 9, 2014

Mark Stehlik

Implementing a generic BST class

- OK, so for a BST node, what do I need to store?
  - Data
  - Left/right links
- And to create a BST?
  - a reference to the root (a TreeNode)
  - and where should the TreeNode class be declared?

Implementing a generic BST class

- So what has to be true of <AnyType> this time?
  - it must be Comparable
  - but since we're constraining AnyType, and **not** the BST class, we say BST<AnyType extends Comparable<AnyType>> instead of *implements*
  - another reason we cannot say *AnyType implements*... is that AnyType could be an interface (which cannot implement anything by definition!)

Implementing a generic BST class

- What methods do we implement on a data structure?
  - constructor
  - isEmpty
  - add
  - traversal [inOrder, the rest are variants]
  - size/count [O(n), for practice]
  - contains/find [O(log n) if tree is balanced)]
  - toString (a rotated tree; not the usual toString!)
  - remove [algorithm only; deferred]

Implementing a generic BST class

- To the code!