

CSSE 230 Day 8

Binary Tree Iterators

After today, you should be able to...

- ... implement `_lazy_` iterators for trees
- ... implement insertion into a BST

Reminders

- ▶ Exam 1 – Day 11: but when and where?
 - Coverage:
 - Everything from reading and lectures, Sessions 1–10
 - Programs through BinaryTrees
 - Homeworks 1–3
 - Allowed resources:
 - Written part: One side of one 8.5 x 11 sheet of paper
 - Programming part:
 - Textbook
 - Eclipse (including programs you wrote in your repos)
 - Course web pages and materials on Moodle
 - Java API documentation
 - A previous 230 Exam 1 is available in Moodle

Exam 1 Possible Topics

- Sessions 1–10, HW1–3, progs through BST
- Written (50–70%):
 - Growable Arrays
 - MCSS
 - big $O/\theta/\Omega$: true/false, using definitions, limits, code analysis
 - Binary search
 - ADT/Collections
 - Choosing an ADT to solve a given problem
 - A little with binary trees (definitions, traversals)
- Programming (30–50%):
 - Implementing one ADT using another ADT

Agenda

- ▶ Binary Tree Iterators
 - Especially (yawn) *lazy* ones
- ▶ BinarySearchTree (BST) insertion

Binary Tree Iterators

What if we want to iterate over the elements in the nodes of the tree one-at-a-time instead of just printing all of them?

Why is the ArrayListIterator an inefficient iterator?

- ▶ Consider a tree with 1 million elements.
- ▶ What is the runtime of iterating over only the first 100 elements?

- ▶ (example on board)

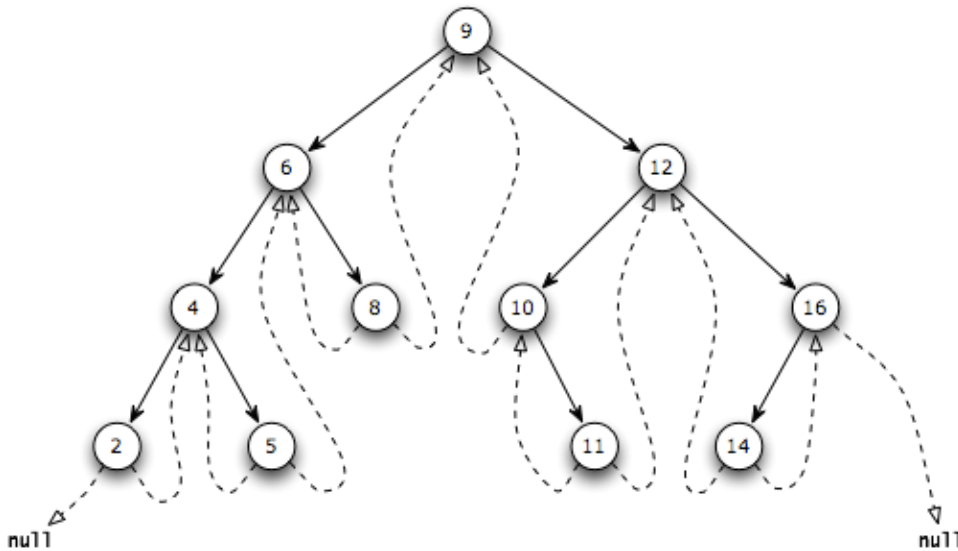
- ▶ To improve efficiency, the iterator should only get as few elements as possible
 - The one time where being lazy has a reward!

Recall the four types of traversals

- ▶ What are they?
- ▶ How would you make a lazy **pre-order** iterator? (brainstorm an algorithm now)
- ▶ What do you need to add to create the other recursive iterators?
- ▶ What about the last iterator?
 - A quick change. Magic? Not really...

Here is a very different alternative to a stack

- ▶ Each node can store pointer to the next and previous nodes in an in-order traversal
- ▶ But must update extra info in constant time as tree changes



Homeworks 5 and 6 will include programming these “threaded binary trees”

Work time

Aim to complete at least Milestone 1
of BinarySearchTrees by next class

We'll start next topic during last 20
min of class

Brainstorm

- ▶ How does one insert into a BST?
- ▶ Rules:
 - Assume you have a BST
 - All elements are Comparable
 - There is only one place to insert the element while keeping the tree a BST
 - Duplicate elements not allowed (we are implementing TreeSet)
- ▶ More on BSTs next class