Homework 4

• write test cases for BoundedAtomicQueue, using MultithreadedTC
• Should be atomic, fair, and the correct size
• You are given a correct implementation, and 4 broken implementations
• Write test cases that distinguish them
• Everything open, no release tests
  – this is a homework, not a project
• Due Friday Nov 16th
Missing office hours

• I’m going to miss almost of my office hours over through Nov 16th
  – various meetings, workshops and trips

• Schedule for coming days
  – Tuesday, Nov 13th 10:45 - noon
  – Thursday, Nov 15th, noon-1:30pm
Upcoming stuff

• WebGoat security homework

• Distributed programming
  – Map/Reduce and Hadoop
Project 3

• Problem with submit server fixed
  – NoSuchMethodException
• Some hints/suggestions
Making sure each node is visited just once

• Absolutely wrong:
  – if (!visited.contains(n)) {
    visited.add(n);
    ... do computations for n;
  }

• The Set.add method returns true if the add method caused something to be added to the set

• If using Java 6, can use Collections.newSetFromMap(new ConcurrentHashMap<Node,Boolean>())
Creating work for a thread pool executor

- Don’t use recursion to explore the graph
- Rather, for each node that should be processed/examined,
  - create a Runnable to process that node
  - give it to a thread pool executor

- If you have a recursive call, you are missing the point of the exercise
Exploring a graph
Can use FutureTask

• This isn’t required, but you can use FutureTask to handle getValue requests
  – create a FutureTask for each node with a callable that invokes the compute method on the node
  – give the FutureTask to a thread pool executor
  – We don’t need to wait for these tasks to complete before returning from explore
Problem

- explore method *must not* return until all nodes have been visited
- No easy way to ask if a thread pool executor is done executing tasks
- How are we going to handle this?
Pop ungraded quiz