CSSE 220

Linked List Implementation

Checkout *LinkedListSimple* project from SVN

Quiz

- Get into pairs
- Look at/run the code in LinkedList.java main
- Draw a box-and-pointer diagram of what's happening in the main code.
- To figure it out, you'll have to look at the LinkedList constructor and addAtBeginning.
- If you've forgotten how to do box-and-pointer diagrams, checkout the handout on Day 5 of the schedule

Q1

Solve the Other Problems in LinkedListSimple

- Look at toString to get an idea of how to do size, then go from there
- They are in approximate difficulty order
- Get help if you get stuck!

Understanding the engineering trade-offs when storing data

DATA STRUCTURES

Data Structures

- Efficient ways to store data **based on how we'll use it**
- The main theme for the rest of the course
- So far we've seen `ArrayLists`
  - Fast addition **to end of list**
  - Fast access to any existing position
  - Slow inserts to and deletes from middle of list

Big-O Notation

- Describes the limiting behavior
  - How slow it can possibly run?
  - Describes the worst case
- Used for Classifying Algorithm Efficiency
- “O” for “Order”
  - O(n) → said as “Order n”
  - O(n^2) → said as “Order n-squared”

Big-O Notation (continued)

- Don't Care About Constants
  - O(2n + 7) → O(n)
- Don't Care About Smaller Powers
  - O(6n^2 + 7n) →O(n^2)
  - Algorithm grows asymptotically no faster than n^2
- If constant value, we say O(1) → "Order 1"
  - O(48) → O(1)

Q2

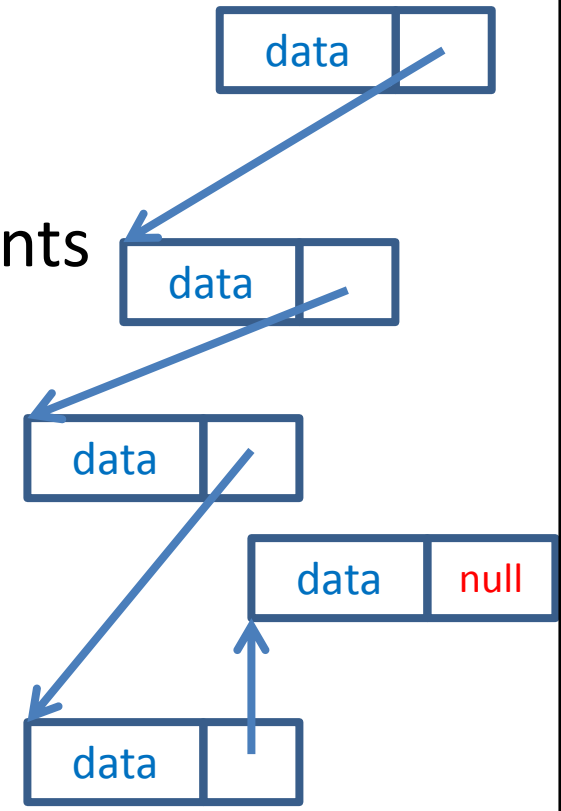
ArrayList Performance (Revisited)

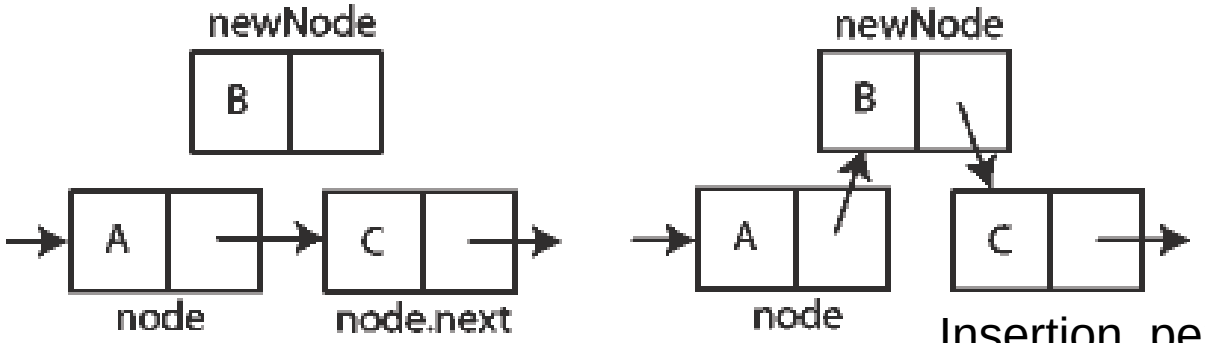
- Fast addition to **end of list**:
  - Fast access to any existing position – O(1) (like array)
  - Keep extra *capacity* for list growth
    - Fast access includes items in capacity not yet filled – O(1)
  - Capacity management is best left for CSSE230
- Slow inserts to and deletes from middle of list
  - Can get to insert/delete location quickly
  - For insert, shift all items right to accommodate -O(n)
  - For delete, shift all items left to fill gap – O(n)

Q3

Another List Data Structure

- What if we have to add/remove data from a list frequently?
- `LinkedLists` support this:
  - Fast insertion and removal of elements
    - Once we know where they go
  - Slow access to arbitrary elements





Insertion, per Wikipedia

Q4-5

LinkedList<E> Methods

- **`void addFirst(E element)`**
- **`void addLast(E element)`**
- **`E getFirst()`**
- **`E getLast()`**
- **`E removeFirst()`**
- **`E removeLast()`**

- What about accessing the middle of the list?
  - **`LinkedList<E> implements Iterable<E>`**

TEAM PROJECT WORK TIME

Software Engineering Techniques

- Pair programming
  - Upcoming assignment *CrazyEights* requires this!
- Version Control
  - How to avoid merge conflicts in SVN

What Is Pair Programming?

- Two programmers work side-by-side at a computer, continuously collaborating on the same design, algorithm, code, and/or test

- Enable the pair to produce higher quality code than that produced by the sum of their individual efforts

Pair Programming

- Working in pairs on a single computer
  - The *driver*, uses the keyboard, talks/thinks out-loud
  - The *navigator*, watches, thinks, comments, and takes notes
  - Person who really understands should start by navigating ☺
- For hard (or new) problems, this technique
  - Reduces number of errors
  - Saves time in the long run

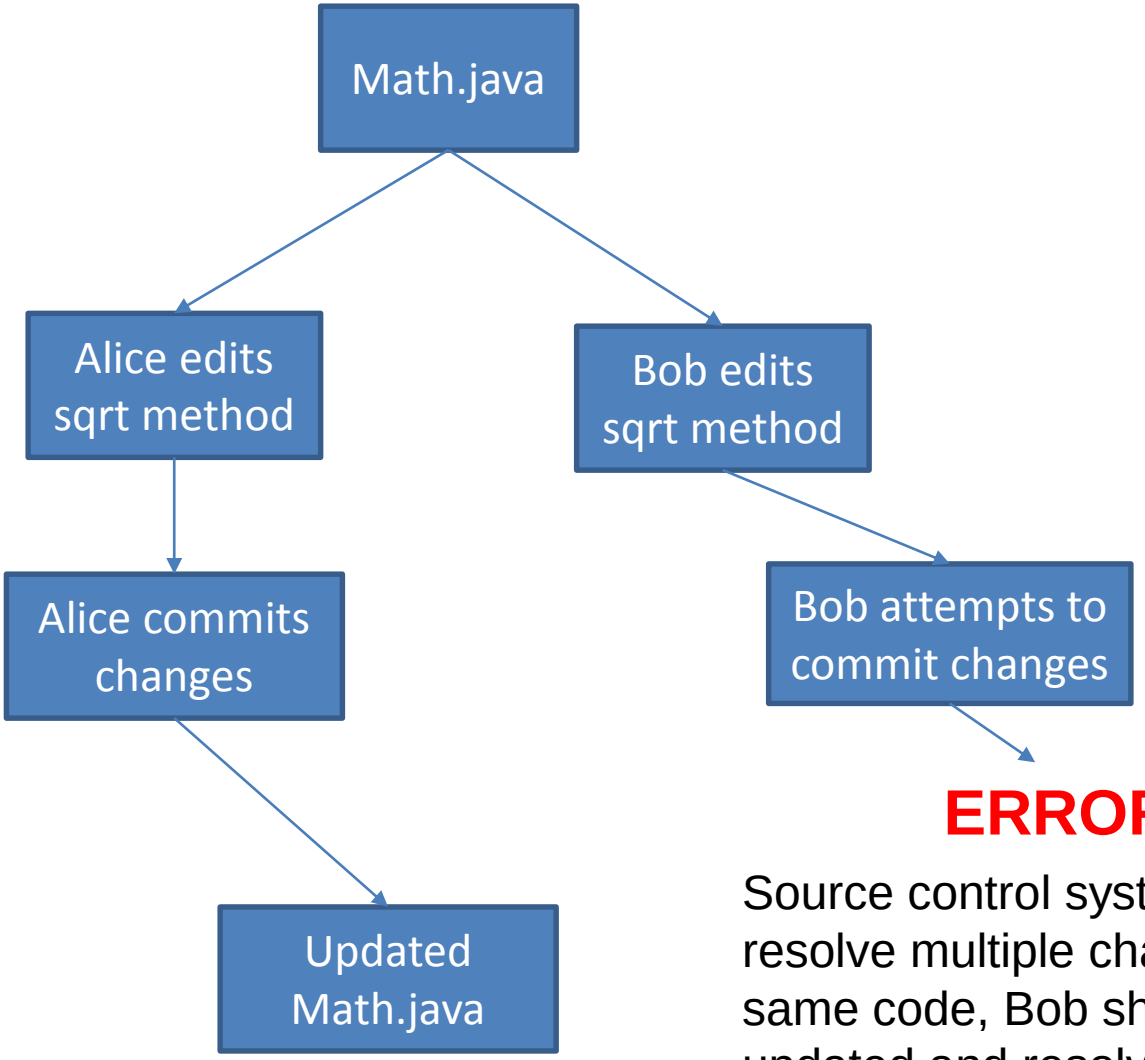
Q1

Pair programming video

- https://www.youtube.com/watch?v=rG_U12uqRhE

SOFTWARE VERSIONS

When Two+ People Edit the Same Code



ERROR

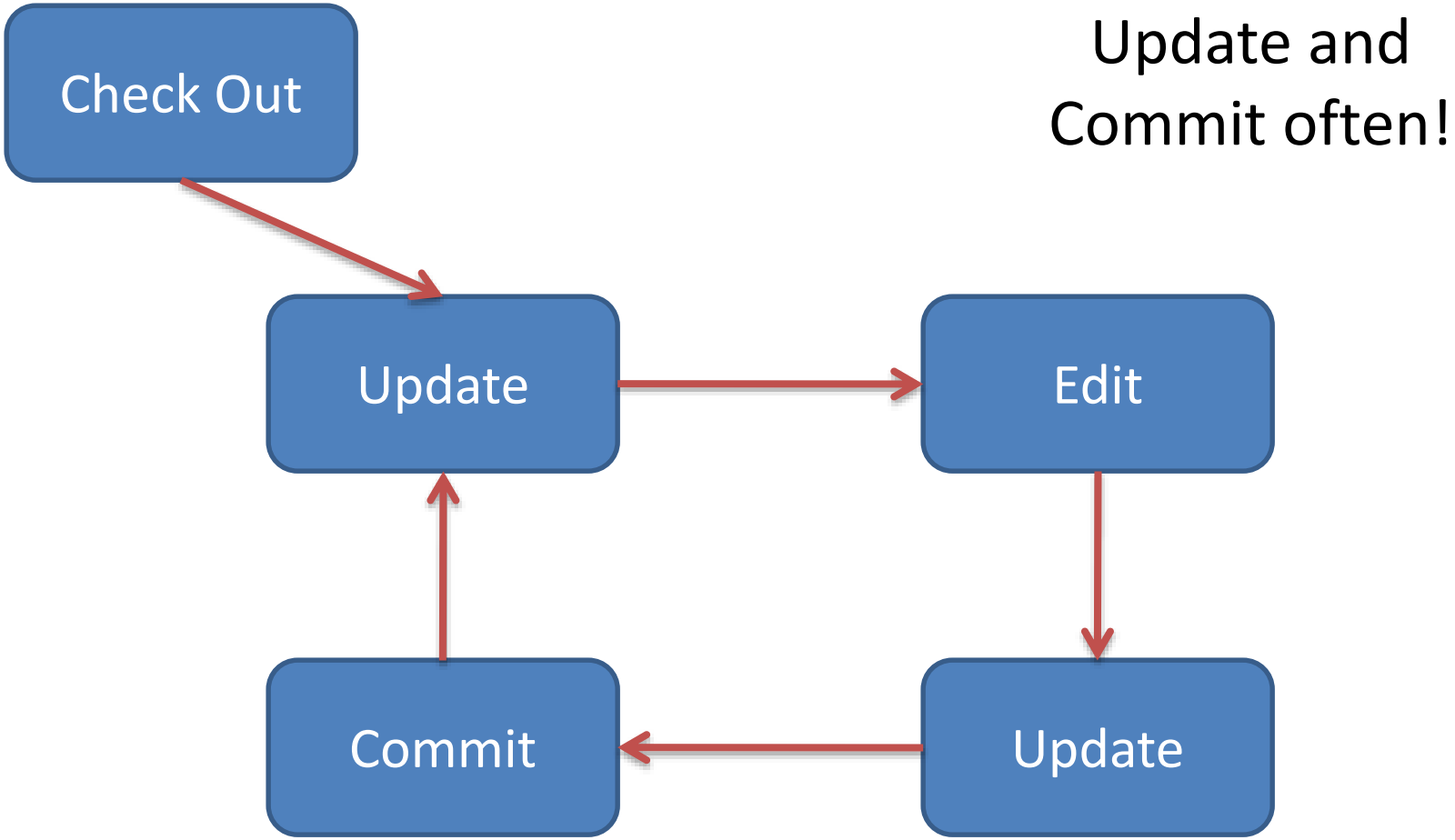
Source control system cannot resolve multiple changes on the same code, Bob should have updated and resolved conflicts before committing.

Team Version Control

- **Version control tracks multiple versions**
  - Enables old versions to be recovered
  - Allows multiple versions to exist simultaneously
- **Always**:
  - **Update before** working
  - **Update again** before committing
  - **Commit often** and with good messages
- **Communicate** with teammates so you don't edit the same code simultaneously
  - Pair programming ameliorates this issue ☺

Q2

Team Version Control



Update and
Commit often!

What if I get a conflict on update?

- If you did an update and now have File.java, File.java.mine, File.java.rN, and File.java.rM (where N and M are integers):
  - YOU HAVE A CONFLICT!
- Eclipse provides tools for resolving conflicts
- Follow the steps in this link to resolve a conflict:
  - http://www.rose-hulman.edu/class/csse/csse221/current/Resources/ResolvingSubversionConflicts.htm