Announcements

• Test 1
  – Closed book, closed notes
  – Except can bring 4 sheets of paper written front/back

• Test 1 Topics
  – Strings, Sets, Maps - hashing, arrays and ArrayLists,
    Classes, Inheritance, Comparable, Files, Scanner,
    Analysis

• Will review in Lab Fri/Mon with old test questions
  – Try test questions before coming to lab, don’t look at
    solutions
  – Practice writing code on paper!

Analysis

• ArrayList – elements not in order
  – Assume n items already in the ArrayList
  – How long does it take to put one new item in?
    • myList.add(value)
    • Worst case?
    • Average case?

• ArrayList – elements in sorted order – maintain
  property
  – Assume n items in ArrayList
  – How long does it take to put one new item in?
    • myList.add(value)
    • Worst case?
    • Average case?

• HashMap
  – Assume n items already in the map
  – How long does it take to put one new item in?
    • myMap.put(key, value)
    • Worst case?
    • Average case?
Analysis

• TreeMap
  – Assume \( n \) items already in the map
  – How long does it take to put one new item in?
    • myMap.put(key, value)
    • Worst case?
    • Average case?

Analysis

• TreeSet
  – Assume \( n \) items already in the set
  – How long does it take to put one new item in?
    • mySet.add(value)
    • Worst case?
    • Average case?

Binary Search

• Given a sorted array of \( n \) names, how do you find a name?

• How does binary search work?
• Can you apply binary search to any array?

• How long does binary search take?
  – Worst case?
  – Average case?

Markov

• Be sure to include analysis
  – Can put in your README file
  – OR include in another document (mention in README file if it is in another document)

• Read the assignment carefully after completing the coding part
Problem

- Given data on books in a library
  - Title, author, year published
  - May have multiple copies of books
- Search for books
  - Sort by title
  - Sort by author
  - Sort by number of copies
  - (we will not focus on how to sort for this problem, just use Collections.sort());

Sort by Title

Sort by Count

How does one sort a type?

- ints, doubles: Compare with <, >, ==
- Strings: how do you compare?
- Book objects: how do you compare them?
Comparable interface

- public interface Comparable
- Classes must implement Comparable interface
- public class Book implements Comparable
- Book must have a compareTo method defined
- Another Example:
  - WordNgram implements Comparable to compare two WordNgrams

How does compareTo work?

- Compares two objects of the same type
- Returns an int
  - returns -1 if first object < second object
  - returns 0 if objects are equivalent
  - returns 1 if first object > second object
- What does compare two Book types mean?
  - By author
  - by title
  - by counts and then author

Classwork today – Amazon Lite

- Library Class
  - Read in the data to create Book object and put them in the library
  - findBooks to return a list of books on some criteria
- Book class
  - Write the method equals to return true if two books have the same author and title
  - Write the method matches – finding a word in a string

Classwork today (more)

- We will compare books several ways
  - 1) We will create Specific Comparator classes and pass them to a sorter
    - TitleComparator (DONE)
    - AuthorComparator
    - CountComparator
  - 2) We will change the Book class to make it Comparable
    - Implement the compareTo method based on titles