CS1004: Intro to CS in Java, Spring 2005

Lecture #21: Algorithms and arrays

Janak J Parekh
janak@cs.columbia.edu

Administrivia

- HW#5 out
  - We'll spend some time talking about it

Array examples, continued

- Before we actually work out the algorithms, how do we structure our programs to work efficiently with them?
- What we'd like to do is to treat the array as a *list*
- What kind of list operations would we like in our array?

Java class model

- 2 or 3 classes
  - A “main” class
  - A class that has the array (“list” class)
  - Possibly, a class that represents the individual items in the list/array
- *All* array manipulation is done through the methods in the second class
  - The main method doesn’t even “see” the array

Sequential Search, analyzed

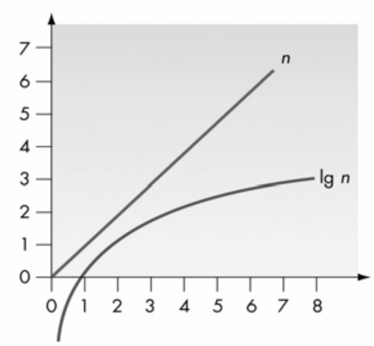
- Comparison of the *NAME* being searched for against a name in the list
  - Central *unit* of work
- For lists with n entries:
  - Best case
    - *NAME* is the first name in the list, 1 comparison
    - O(1)
  - Worst case
    - *NAME* is the last name in the list, or not in list
    - n comparisons, or O(n)
  - Average case
    - Roughly n/2 comparisons, or O(n)

Sequential Search (continued)

- Space efficiency
  - Uses essentially no more memory storage than original input requires
  - Very space-efficient
- But… is there a faster way to search through a list?

Binary Search

- Given ordered data,
  - Search for *NAME* by comparing to middle element
  - If not a match, restrict search to either lower or upper half only
  - Each pass eliminates half the data
- Efficiency
  - Best case
    - 1 comparison: O(1)
  - Worst case
    - lg n comparisons: O(lg n)... *What's lg n?*
    - Fundamental idea: given N steps, how many elements can we process?



A Comparison of n and lg n (S/G, pg. 109)

Largest number

- Goal: given an array of N items, find the largest one
- How much additional space do we need to store?
- How long does it take for this algorithm to run?

Sorting

- What if we want to sort the numbers in a list?
- There are number of algorithms; book describes selection sort, but we'll also go over bubble sort quickly.
- Let's begin!

L/L Chap 5.9-5.12

- Basically the same GUI concepts covered in chapter 4, but with loops and conditionals
- "Read-only" – take a look through in your spare time, understand the concepts
- We may have GUI programming on HW#6, but there won't be on the final

Next steps

- We finally have a good idea of algorithms and ways to tell Java to structure data for them
- How do we choose the appropriate structure?
  - Either have your instructor tell you to, or;
  - Learn it yourself
- We'll start exploring design methodologies soon, but this is a lifelong learning process
- Next: discuss HW#5

Next time

- Start OO