Administrivia

- The bookstore definitely has books…
- Bug in compareTo example on page 107
  - Should be “s1.compareTo(s2)” in the table header
- HW1 updates

Agenda

- Finish writing out some Java list code
- Basic big-Oh notation
- Begin sorting

Lists

- Ordered Insert
  - Book has a cleverer technique; see page 60
  - Once you find the Insert point, work from the bottom up
- Ordered Find
  - Book page 57; very similar to what I did, but some subtle differences

Costs

- How much do each of the previous entries cost in the worst case?
  - Most are linear, some are unit
- Binary search is special – it’s better than linear time
  - Divide the range by half until too small to divide further == # of comparisons needed
  - Reverse: what’s the range that can be covered with \( n \) steps? (Book page 63)
    - i.e., \( r = 2^s \)
    - What’s this expressed as in terms of \( s \)?
      - \( s = \log_2 r \)
      - Algorithm grows logarithmically

Formalizing costs

- Terminology differs based on details; we’ll go light
- Time to insert one element is some constant \( K \)
  - e.g., \( T(N) = K \)
- Time to search for an element is \( T(N) = K \times N \)
- “Big-Oh Notation”: upper-bound on worst-case time
  - We drop the constant \( K \) – for sufficiently large \( N \), the constant is unimportant
  - The idea of doubling your computer’s speed is embedded in \( K \)
  - \( T(N) = O(N) \), for example

Examples of costs

- For lists using arrays?
  - Linear search: \( O(N) \)
  - Etc.
  - Draw a graph of the comparative costs, page 72
- What are bad about arrays?
– Slow search in unordered, slow insert in ordered – can we speed both? Yes
– Fixed size
– But it’s easy
– You can write your own Vector if you want

8 ▶️ Sorts
• Applets!
• Bubble (p. 85)
  – Sort pairwise repeatedly
  – Biggest placed each time
• Selection (p. 89)
  – Search for smallest, swap with first
  – Search for smallest, swap with second
• Insertion (p. 95)
  – Take the next one, and put it into the existing sorted subset
• All O(n²)
  – But they’re not the exact same performance
  • Let’s write out a little bit of pseudocode for each

9 ▶️ Next time…
• Finish sorts
• Stacks