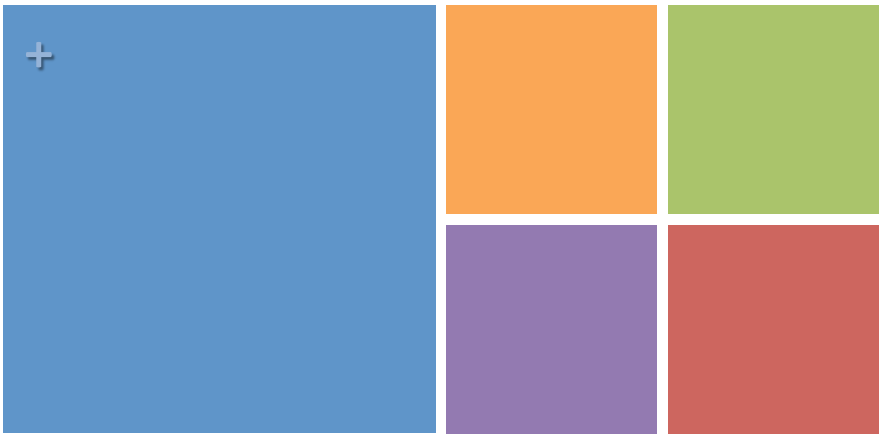

Lecture 5: ArrayList implementation & Complexity

Today

- Reading
  - JS Ch. 3 (Vectors) and Ch. 4 (Generics)
- Objectives
  - Implementation of ArrayList
  - Introduction to Big-O notation (Complexity)
- Announcements
  - 4 guest lectures in February

Back to ArrayList

- Our first example of how to analyze the complexity of a data structure
- See ArrayIndexList<E>
  - Similar to ArrayList
  - Instance variables:
    - elts: array instance variable,
    - eltsFilled: number of slots filled.
- Some operations very cheap:
  - size, isEmpty, get, set take constant time (no search)
- Others more expensive

Adding elements

- Easy if there is space:
  - At end, just add it
  - If before end, move all elements at i and beyond to right before inserting
  - *Delete similar*
- What should we do if there is no space?
- How expensive is this solution?

Complexity of Operations

- Count number of compares and/or moves to accomplish operation.
- Rather than keeping an exact count of operations, use *order of magnitude* count of complexity.
- Ignore differences which are constant
  - n and n/2 have the same order of magnitude.
  - 2 n^2 and 1000 n^2 have the same order of magnitude

Order of Magnitude (on board)

Complexity

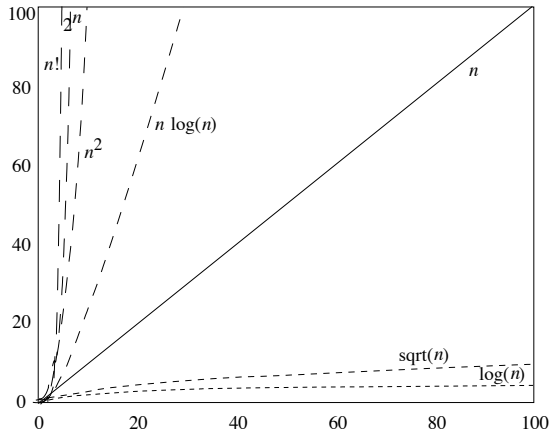

Figure 5.3 Long-range trends of common curves. Compare with Figure 5.2.