Lecture 6: Complexity

CS 62 - Spring 2016
Michael J. Bannister

Lab This Week

- Timing ArrayList operations
  - Encourage working in pairs
  - Stopwatch class: start(), stop(), getTime(), reset()
- Java has Just-In-Time compiler
  - Must “warm-up” before you get accurate timing
  - What can mess up timing?
- Uses Vector from Java libraries because can change way it increases in size.

Programming Assignment This Week

- Weak AI/Natural Language Processing:
  - Generate text by building frequency lists based on pairs of words. ArrayList of Associations of String (words) and Integer (count of that word).

Complexity

- Count the number of elementary operations used to perform a task (i.e., a method).
- Elementary Operations:
  - Read/Write
  - Arithmetic

Adding to ArrayList

- Suppose n elements in ArrayList and we add 1 element.
- If space:
  - Add to end is a constant number of ops
  - Add to beginning is ~n copy ops
- If not space,
  - What is cost of ensureCapacity?
  - ~n copy ops because n elements in array

Multiple Adds

- What if only increase in size by 1 each time?
  - Adding n elements one at a time to end
    - Total cost is 1+2+3+...+(n-1) = n(n-1)/2 copy ops
- What if double in size each time?
  - Suppose add n new elements to end
    - Total cost is 1+2+4+...+n/2 = n-1 copy operations

O-Notation

- Definition: We say that *g(n)* is *O(f(n))* if there exist two constants C and k such that $|g(n)| <= C\ |f(n)|$ for all n > k.
- Used to measure time and space complexity of algorithms on data structures of size n.
- Ignores constants and lower order terms

ArrayList Ops

- Worst case
  - O(1): size, isEmpty, get, set
  - O(n): remove, add
- Amortized: O(1) for all operations