

EIGHTH (AND LAST!) QUIZ

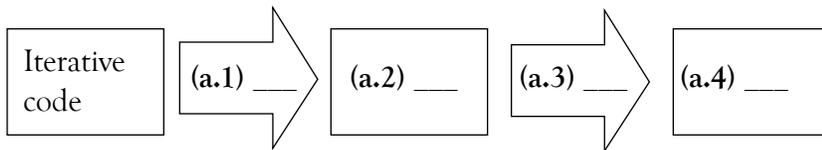
You have 15 minutes from the start of class to complete this quiz. Give partial answers if you can't give complete ones. Read the questions with care; work with deliberate speed. Don't give us more than we ask for. The usual instructions apply. Good luck!

Problem 1 (10 points)

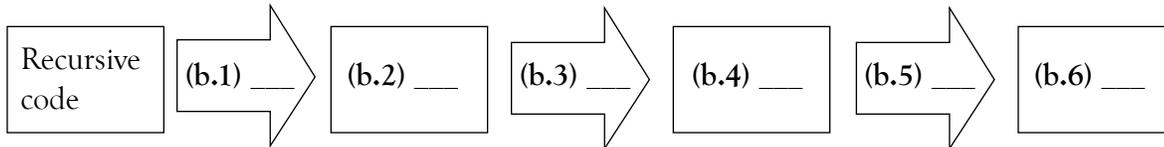
These are some steps and results involved in determining the performance of a given segment of code:

- A. O-notation
- B. Polynomial
- C. Add the operations you're counting; multiply loop bodies by number of times through loop, ...
- D. Recurrence relation
- E. Remove lower-order terms, remove lower-order O-notations, remove constants
- F. Count operations for the base case; count operations for the recursive case (usually in terms of the next smaller case)
- G. Solve the recurrence

(a) (4 points) Describe the process of deriving an O-notation from iterative code by filling in each blank ((a.1) through (a.4)) in the diagram below with a letter (A through G) from the list above; you won't use every letter. (Hint: The arrows contain verbs or verb phrases; the boxes contain nouns or noun phrases.)



(b) (6 points) Describe the process of deriving an O-notation from recursive code by filling in each blank ((b.1) through (b.6)) in the diagram below with a letter (A through G) from the list above; you won't use every letter. (Hint: The arrows contain verbs or verb phrases; the boxes contain nouns or noun phrases.)



Problem 2 (5 points)

Which of the following are accurate statements about sorting algorithms and the film *Sorting out Sorting*? (Answer true or false.)

- _____ All the algorithms shown in the film were either $O(n \log n)$ or $O(n^2)$.
- _____ We sometimes distinguish comparisons from data movements because comparisons are slower.
- _____ Using a tree-based sort guarantees you at least $O(n \log n)$ performance.
- _____ Exchange sorts swap values' positions to bring them closer to the correct order.
- _____ With some $O(n^2)$ sorting algorithms, as n gets larger, the amount of time required levels off.

Problem 3 (10 points)

These are some programming languages we discussed in class: Fortran, Cobol, Algol (58/60), Lisp, Basic, PL/I, Simula, Algol 68, Pascal, C, Scheme, Ada, Common Lisp, C++, Java, Python, C#, F#.

(a) (1 point) List at least two languages from the list above in which functional programming is required, or at least a common programming idiom.

(b) (1 point) List at least three languages from the list above that were designed to support object-oriented programming.

(c) (1 point) List at least three languages from the list above that were developed before 1960.

(d) (1 point) List at least one language from the list above that was developed with simplicity or minimalism as a major design goal.

(e) (3 points) Write the name of the appropriate language from the list above next to its developer's name below. For full credit, you only need three correct matches; there is no penalty for wrong guesses. You may use the same language name more than once. Some names may match more than one language (but only one correctly matching language is required for credit).

_____ John Backus	_____ Grace Hopper	_____ Dennis Ritchie
_____ Noam Chomsky	_____ John Kemeny	_____ Guy Steele
_____ Alain Colmerauer	_____ John McCarthy	_____ Bjarne Stroustrup
_____ Matthias Felleisen	_____ Donald Knuth	_____ Guido van Rossum
_____ James Gosling	_____ Thomas Kurtz	_____ Niklaus Wirth

(f) (3 points) Write the name of the appropriate language next to each item below. For full credit, you only need to match three items correctly; there is no penalty for wrong guesses. You may use the same language name more than once. Some items may match more than one language (but only one correctly matching language is required for credit).

_____ Adopted syntax similar to C and C++ but a different type-safety philosophy

_____ The first language designed for interactive programming

_____ Programs look superficially like English

_____ Combined features of Fortran, Cobol, and Algol

_____ A scripting language, convenient for “gluing” other applications together