1 Course Link

You can access this course at [this link] if it is available and you are enrolled in it.

2 Catalog Description

Introduces formal techniques to support the design and analysis of algorithms, focusing on both the underlying mathematical theory and practical considerations of efficiency. Topics include asymptotic complexity bounds, techniques of analysis, and algorithmic strategies.

3 Course Goals

1. Become conversant with the topics and issues surrounding algorithms and complexity. These include (but are not limited to):
   - Basic algorithms analysis: Asymptotic analysis of upper and average complexity bounds;
   - Best, average, and worst case behaviors;
   - Big-Oh, little-Oh, Big-Omega, and Big-Theta notation;
   - Standard complexity classes;
   - Empirical measurements of performance; time and space tradeoffs in algorithms;
   - Using recurrence relations to analyze recursive algorithms;
   - Fundamental algorithmic strategies: brute-force; greedy; divide and conquer; decrease and conquer; backtracking;
   - Graph and tree algorithms: depth-and-breadth-first traversals; shortest-path (Dijkstra’s and Floyd’s algorithms); minimum spanning tree (Prim’s and Kruskal’s algorithms); topological sort.

2. Learn the techniques (i.e., acquire the “tools”).
   - Analyze and compare algorithms using Big-Oh, Big-Omega, and Big-Theta.
   - Describe and implement in a high-level language (C++) some or all of the following algorithmic techniques: Brute Force, Divide/Decrease/Transform-and-Conquer, Greedy, Dynamic Programming, Iterative Improvement and Backtracking;

4 Prerequisites

4.1 Courses

- CS 235 Data Structures
- CS 237 Discrete Mathematics I
4.2 Knowledge

- Object oriented C++
- Basic data structures (lists, trees, graphs, etc.)
- Summation notation (Σ)
- Recurrence relations
- Matrices
- Limits
- Logarithms
- Proofs

5 Course Materials

Required Text:
Introduction to the Design & Analysis of Algorithms
Anany Levitin
ISBN: 9780132316811

6 Requirements

You are required to...

- attend class, as attendance and participation factor into your grade.
- read assigned portions of the course materials before the class meeting when they will be discussed.
- take quizzes to make sure your reading is effective.
- take tests to reinforce the concepts you have learned.
- do homework assignments to enhance your understanding of selected topics.
- take a comprehensive final exam.

7 Assignments

Homework assignments come in two flavors: Exercises and Explorations.

7.1 Exercises

Exercises are homework assignments meant to help reinforce the concepts we discuss in class. These consist mostly of exercises from the book that will be due a day or two after we discuss their section’s material. We may go over a few of them in class. They will be graded by you after a grading key is made available.

There will also be exercises in the form of individual and group learning activities that will require either a small amount of preparation before class, or participation during class, or both. These will also be self-graded.
7.2 Explorations

Explorations are programming or project-type assignments that allow you to explore some of the topics we will investigate. They are designed to deepen your understanding of key course concepts, and entail writing papers and programs. All papers will meet college level writing standards and will be graded on such criteria as organization, quality of information, grammar and spelling, and how responsive they are to their requirements. All programming will be done in C++ and must compile and run correctly on the Linux machines in AUS 213.

7.3 Late Policy

Work is due on the day indicated in the schedule. Late work is accepted only if the reason is extraordinary, and acceptance is reached through private and prolonged negotiation. And you must come talk to me in person — no emails, phone calls, nor texts.

8 Assessments

Assessments come in two flavors: Quizzes and Tests.

There will be frequent preparation-assessment (take-before-class mostly, but some take-during-class) quizzes. These consist of multiple choice, multiple answer, true/false, fill in the blank or matching type questions for the online quizzes, and write-on for the in-class quizzes. These quizzes are open book, open notes, but you should try to take them without using book or notes (or google, etc.)

Five tests will be given as outlined in the schedule. These will be closed book, closed notes, (closed google, closed wikipedia, etc.) and will be proctored by the Testing Center.

Lastly, a comprehensive final exam will be given at the end of the semester, per the schedule. This exam will be administered in the classroom or the Linux lab, NOT in the testing center.

9 Grading

- Assignments: 50%
  - Exercises: 15%
  - Explorations: 35%
- Assessments: 50%
  - Quizzes: 10%
  - Tests: 20%
  - Final: 20%

Your weighted percentage will determine your final grade as output from the following function:

```lisp
(defun convert-to-letter-grade(weighted-percentage)  
  (let* ((number (ceiling weighted-percentage))  
     (tensDigit (/ number 10))  
     (onesDigit (mod number 10))  
     (index (min (max (- tensDigit 5) 0) 4))  
     (letter (substring "FDCBA" index (+ index 1))))  
     (sign (if (<= onesDigit 2) "-" (if (>= onesDigit 7) "+"))))  
  (concat letter (if (and (< number 95) (>= number 60)) sign))  
)
```