CS 330
ALGORITHM ANALYSIS

Fall, 2017

Prerequisites

CS 280 - Data Structures
CS 225 - Advanced C/C++
MAT 200 - Calculus and Analytic Geometry II or
MAT 230 - Vector Calculus II

General Information

Class Schedule:	Mondays 10:30am-11:50am Wednesdays 10:30am-11:50am
Class room:	MICHELANGELO
Professor:	Eva Iwer
Email:	eva.iwer@digipen.edu
Phone number:	*x*5089
Class web page:	https://distance.digipen.edu/2016-fall/my/ http://faculty.digipen.edu/~eva.iwer https://pontus.digipen.edu/
Office Hours:	by appointment

Description

This course provides students with an introduction to the analysis of algorithms, specifically proving their correctness and making a statement about their efficiency. Topics for discussion may include loop invariants, strong mathematical induction and recursion, asymptotic notation, recurrence relations, and generating functions. Students examine examples of algorithm analysis from searching and sorting algorithms.

Course Objectives and Learning Outcomes

This course is designed to promote individual learning and analytical thinking skills. It combines lecture, reading and projects implementation. Upon successful completion students should

- Be able to prove correctness of algorithms
- Have deep understanding of differences and advantages of iterative and recursive algorithms

- Have a knowledge base of existing algorithms
- Understand speed vs. space trade-off, importance of data-structures, and data preprocessing
- Be able to design new algorithms using the ideas from class

Academic Integrity Policy

Academic dishonesty in any form will not be tolerated in this course. Cheating, copying, plagiarizing, or any other form of academic dishonesty (including doing someone else's individual assignments) will result in, at the extreme minimum, a zero on the assignment in question, and could result in a failing grade in the course or even expulsion from DigiPen.

Disability Support Services

If students have disabilities and will need formal accommodations in order to fully participate or effectively demonstrate learning in this class, they should contact the Disability Support Services Office at (425)629-5015 or dss[at]digipen[dot]edu. The DSS Office welcomes the opportunity to meet with students to discuss how the accommodations will be implemented. Also, if you may need assistance in the event of an evacuation, please let the instructor know.

Course Materials

Textbooks

Introduction to the Design and Analysis of Algorithms by Anany Levitin, Publisher Pearson Education Limited, ISBN 1292014113, 9781292014111, 9780201743951

Optional Textbooks

Additional readings may be assigned and be handed out and made available on Moodle

Assessment

Grading Policy

Grades will be derived from homework assignments and exams.

Grade breakdown:

- Assignments: 40%
- Quiz: 10%
- Midterm exam: 20%
- Final exam: 30%

You must receive an average score of 60% on both the midterm and final exams combined to pass this course, regardless of your homework/quiz scores.

Letter grade distribution:

X %	Grade
X ≥ 93	A
90 ≤ x < 93	A -
87 ≤ x < 90	B +
83 ≤ x < 87	B
80 ≤ x < 83	B -
77 ≤ x < 80	C +
73 ≤ x < 77	C
70 ≤ x < 73	C -
60 ≤ x < 70	D
X < 60	F

Mechanisms and Procedures

Homework

You will be given no less than 14 days to complete each assignment. This gives you adequate time to manage your workload. Deadline is 11:59 PM (PST) 2 weeks after getting the assignment. This means if assignment starts at 11-01 then the deadline is 11-15 11:59 PM (PST). The amount of time actually required to complete an assignment is much less than the time allotted and is generally between 5 and 20 hours. You should plan on devoting 6 hours per week to this course.

Assignments will be accepted once and revisions will not be allowed.

To assure proper credit for your work, please indicate the following information on each assignment submission:

- Course Number
- Your Name
- Assignment Number

The assignments must be submitted electronically through Moddle (distance.digipen.edu) if not otherwise declared.

Programming Assignments

Programming assignments will use the C++ programming language. More specifically, all programs must adhere to Standard C++. Assignments will be graded using the GNU C++ compiler version 5.4, but the assignments will also need to be compiled and tested with other compilers (e.g. Microsoft) and will be specified in the assignment. You are encouraged to build and run your programs with many compilers, since this is the only way to help ensure that your code is legal and robust. Any submissions that do not compile cleanly will not be accepted and will result in a grade of 0. Additional detailed instructions will be provided with each assignment. This usually includes a handout as well as supplemental material available on the web site.

Each homework assignment that is to be submitted will specify the time/date that it is due. Late assignments will not be accepted. There is more than enough time in your schedule to complete all homework assignments on time. Of course, if you wait until the due date/time is near to begin your homework, you may not finish it on time. Time management is your responsibility.

Exams

All exams are closed book. One sheet of notes is permitted for an exam, as this course is about understanding, not memorization. This sheet of notes must be handwritten by you, and no larger than a 'normal' (8½" x 11") piece of paper. Front and back of the page may be used. The midterm will cover all material taught up to the exam date. Any material covered in the course is valid for testing; including concepts covered in lecture, homework, or other communications and/or assigned work.

Make-up exams are only available for excused absences.

Late Policy

Assignments will NOT be accepted after the submission deadline.

Course Outline and Tentative Dates

The following topics and dates are subject to change at any time.

WEEK	TOPIC	DETAILS
1 - 3	Background and Fundamentals	• logarithms, big-O notation • iterative vs. recursive algorithms • algorithms for generating permutations, subsets, and combinations • Gray coding - discussion of locality of reference List of problems to be used in class • sorting a collection • maximum, minimum elements, other statistics • searching in a collection • multiplication of large numbers • string matching • closest pair of points in a plain • convex hull of a set of points • longest increasing subsequence • longest common substring of 2 strings (LCS) • knapsack • shortest path between 2 vertices of a graph • all-pairs shortest path • minimum spanning tree of a graph (MST) • connected components (bidirectional graph) • transitive closure (similar to above, but directional graph) • 3-SAT Analysis of algorithms: • proving correctness. Examples: fast exponentiation, remainder. • time complexity (worst, best, average). Example - fast exponentiation. • time complexity - recursive algorithms. Solving recurrences. • case study - Fibonacci numbers. P vs. NP
4 - 5	Brute-force algorithms	• sorting - snail, selection, and bubble sorts • searching - sequential search • string matching • convex hull

		• closest-pair • knapsack Clever brute-force - backtracking: generating tuples
6	MIDTERM	
7-9	Divide-and-conquer	• general structure (pre-processing, recursive call(s), post-processing) • sorting - insertion sort (divide into n-1 and 1 piles) o mergesort o quicksort o binary search (application to debugging) • convex hull • closest-pair knapsack - cannot be (reasonably) solved using divide-and-conquer. Discussion.
10-11	Greedy algorithms	• definition and discussion of sub-optimality • shortest path (Dijkstra) MST (Prim, Kruskal)
12	Dynamic programming	• introductory example - computing Fibonacci numbers • LCS • knapsack • transitive closure - Warshall's algorithm • all-pairs - Floyd's algorithms comparison with recursive algorithms and memory functions
13	Iterative improvement algorithms	• definition and discussion of convergence • solving non-linear equations - Newton's method • Revisiting DFS and BFS: may be seen as brute-force or reduction-by-1

Additional Information

Last Day to Withdraw

In order to withdraw from a course it is not sufficient simply to stop attending class or to inform the instructor. In accordance with the policy, contact your advisor or the Registrar to begin the withdrawal process. The last day for withdrawal from this course is cited in the official catalog.}

Academic Support Center

The Academic Support Center, located on the 2nd floor next to Gibran and Edison, offers free tutoring sessions for select 100 and 200 level courses. Tutors are trained to enhance the understanding of core course concepts, answer questions, and assist with exam preparation. Drop-in tutoring is available throughout the day or students can schedule a drop-in appointment. For any additional questions regarding Tutoring Services, please contact studentsuccess@digipen.edu.

Cell Phone Policy

Cell phones are to be silenced and unanswered in class without express permission of the instructor. This includes calls, text messages, or any other functionality your cell phone might have.

Classroom Policy

- No food.
- Drinks are allowed, unless prohibited by School policies.
- No loud noises.
- No phone calls/ ringing.
- Laptops are allowed if used to display lecture material.

Classroom Procedures:

- Come to class every day. If you cannot attend class due to extenuating circumstances, contact the instructor via email and arrangements can be made.
- Please utilize the message boards on distance for all questions you have about implementation details for the assignments; remember, someone else probably has the same question that you do so posting to the forums is much more efficient in ensuring that everyone's questions are answered.
- All students should retain a copy of all of their coursework until the end of the semester.
- If you feel that your assignment or exam was not graded correctly please contact the instructor via email, during office hours, or immediately before/after class (time permitting).

Use of course materials