CS 330/535: Algorithm Analysis

Prerequisites

CS 225, CS 280, MAT 200 or MAT 230

General Information

- Instructor: Dmitri Volper
- Class schedule: 9-10:40 T/Th (Blanc)
- Office Phone Number: 7017
- Email address: dvolper at digipen.edu
- Home page (DigiPen): `faculty.digipen.edu/~dvolper`
- Office hours: T/Th 11-12 or 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 will examine examples of algorithm analysis from searching and sorting algorithms. Second part of the course will concentrate on classification of algorithms and building a strong knowledge base of existing algorithms.

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

- examine run-time complexity of a program based on which algorithm and data structure are used
- compare and contrast properties of the algorithm and data structure for optimal usage
- select or design an appropriate algorithm for a computing problem
- understand the tradeoffs between different types of algorithms and data structures

Academic Activity Documentation and Census

As of Fall 2021, DigiPen Institute of Technology is a non-attendance taking institution. Federal regulations require our institute to document that each student has begun attendance in all enrolled courses. This documentation will be recorded in the course website, in Moodle.

Every course has required academic activities. Required academic activities during the first two weeks will be used to form a census. A student with no recorded academic activity in a course at the end of the second week of the semester will be withdrawn from the course.

Note: Each student will have at least two (2) opportunities to provide documented proof of activity, but one (1) instance of academic activity is sufficient to prove academic activity and avoid withdrawal. Students who cannot submit any academic activity in the first two weeks of class cannot be excused by the instructor or the Office

of Disability Support Services. Those students will have to work with the Office of the Registrar regarding their automatic withdrawal from the course.

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.

Religious Accommodation

DigiPen Institute of Technology provides reasonable accommodations to students who may be absent from activities or incur significant hardship due to religious holidays or observances. These holidays or observances must be part of a religious denomination, church, or religious organization, and the course instructor must be notified in writing during the first two weeks of the course. The institute's policy for grievances is published in the course catalog.

AI tools

In this course, the use of Large Language Models (LLMs), such as ChatGPT, is permitted as a tool for assisting in the understanding of course material. Students are encouraged to use LLMs for clarifying concepts, explaining difficult topics, and engaging in discussions that help enhance their learning experience. However, it is important to note that LLMs should not be used to complete tests, assignments, or any other form of evaluative work. Submitting work generated by an LLM as your own is considered academic dishonesty and violates the principles of integrity expected in this course. Students are expected to demonstrate their understanding and skills independently.

Course Materials

Textbooks

- *Algorithms* by Jeff Erickson `http://jeffe.cs.illinois.edu/teaching/algorithms/`
- *Data Structures and Algorithm Analysis* by Clifford A. Shaffer `https://people.cs.vt.edu/shaffer/Book/C++3elatest.pdf`

References (Optional)

Assessment

Grading Policy - underghaduate

- 30% – 5 programming assignments
- 10% – 6 programming labs

- 10% – 6 written homeworks
- 25% – 1 midterm exam
- 25% – 1 final exam

Grading Policy - graduate

- 20% – 5 programming assignments
- 10% – 1 presentation
- 10% – 4 programming labs
- 10% – 6 written homeworks
- 25% – 1 midterm exam
- 25% – 1 final exam

Grading scale

- $x \geq 93$ is Excellent (A)
- $90 \leq x < 93$ is Excellent (A-)
- $87 \leq x < 90$ is Good (B+)
- $83 \leq x < 87$ is Good (B)
- $80 \leq x < 83$ is Good (B-)
- $77 \leq x < 80$ is Fair (C+)
- $73 \leq x < 77$ is Fair (C)
- $70 \leq x < 73$ is Fair (C-)
- $60 \leq x < 70$ is Poor (D)
- $x < 60$ is Fail (F)

Mechanisms and Procedures

Submitting Homework/Assignments/Labs

The source files must be submitted electronically through the course submission page - use your DigiPen login and student number to login.

`https://pontus.digipen.edu/cgi-bin/submission.cgi`

There will be several programming assignments during the semester. Expect that during any week there will be an assignment that you can work on. The assignments' due dates will be distributed throughout the semester as evenly as possible, but sometimes there may be two assignments due in two consecutive weeks. You will always have at least one week for any given assignment without overlaps with any other assignment. Each assignment is supposed

to take 4 to 6 hours (including downloading the handout and driver, understanding the requirements, drawing, thinking, actual coding, testing, debugging (repeat the last three steps required number of times), commenting, and submitting). In some cases the time may be doubled, tripled, etc. due to your error in understanding of a requirement, some weird debugging problem, you being tired, upset, etc. Plus remember that there will be more assignments from other classes due at about the same time. The best way of dealing with this is to start each assignment (project/homework) **as soon as possible**.

The above is also true for the labs, but labs will be taking significantly less time (1 to 2 hours) because of a smaller scope and the documentation not being required.

Late Policy

There will be a default late submission policy for the programming assignments and labs – one (1) percent penalty for each hour past the due time for 48 hours. Apart from that no late submissions will be allowed. The only reason I will accept late homework without a penalty is if you had an emergency. Notify me as soon as possible if this is the case and we will negotiate the extension. In any case – I expect you to contact me before the assignment's due time (unless the nature of the emergency prohibits you from doing so). That means that in general I will not consider requests for extension if the due date is already in the past.

What is an *emergency*? There is no formal definition, but approximately – any situation that was **not under your control** and was **not known beforehand** that **prohibits you from working** on a given test for **an extended period of time**.

Examples of emergencies: unexpected situation involving your family that requires your attention, sprained arm, hand, back, migraine or other head-related pains (doctor's notice is required), power outage for more than 6 hours for example due to snow or wind storm.

Examples of non-emergencies: travel for a wedding (since it is known in advance you should plan your time accordingly), too much work because of other classes (plan ahead, start early), no power or internet for an hour.

Examples of 'gray area': loss of power (was it more than 2 hours?, did you have to wait till the last hour to work on the test?, can you go to campus?), loss of internet (all of the previous questions plus did you consider tethering?), sprained leg, foot (technically you do not need it for typing). I will be reasonable and treat most of the gray area cases as emergencies, I also understand that a pain in a foot may be distracting so you cannot code, that you will not be going to see a doctor if it is a simple cold or a random headache, and not require a doctor's note. But I will only do it once per student, if the problem persists – you need to see a doctor or change your attitude.

Course Outline and Tentative Dates

- week 1: Intro to course, big-O
- week 2: Brute-force/combinatorics
- week 3: Backtracking
- week 4: Divide and Conquer
- week 5: Midterm.
- week 6: Greedy
- week 7: Dynamic programming
- week 8: Iterative improvement algorithms
- week 9: Event driven design

- week 10: Tradeoffs
- week 11: Computational complexity
- week 13: Final Exam