Syllabus (Spring 2024)

Course Information

Catalog Number	56:198:501 (also cross-listed as 56:219:501 and 56:121:531)
Instructor	Sunil Shende
Schedule	**W from 6pm to 8:40pm**
Classroom	**BSB-116** Business and Science Building
Office Hours	MW on Zoom **from 3pm to 4pm**
Email	shende AT camden DOT rutgers DOT edu

Learning Goals

1. Design, implement and understand operational details behind basic data structures such as linked lists, binary search trees, hash tables and skip-lists.
2. Implement and analyze searching and sorting with appropriate data structures.
3. Understand and analyze three fundamental algorithm design paradigms: **divide-and-conquer**, **greedy** and **dynamic programming**.
4. Develop fluency in basic mathematical tools and techniques needed to analyze algorithms, check for their correctness, and establish their complexity using asymptotic notation.

Reference Material

We will not have an assigned textbook as such but I will draw upon several excellent sources for material on data structures and algorithms.

1. Algorithms Unlimited
2. Problem Solving with Algorithms and Data Structures Using Python (3rd Edition)
3. VisuAlgo

Links to relevant topics from these sources will be provided on the Rutgers **Canvas** site for the course.

Additional resources include the following excellent books and websites:

- **Algorithms Unplugged**, by Thomas Cormen; The MIT Press (2013).
- **Algorithms** by Dasgupta, Papadimitriou and Vazirani; McGraw-Hill Education (2006).
- **Introduction to Algorithms (3rd edition)** by Cormen, Leiserson, Rivest and Stein; The MIT Press (2009).
- MIT's 6.006 Course (Fall 2011) playlist is based on the textbook by Cormen et al. listed above.
- Open Data Structures

Logistics

This course is now a core course in the Computer Science MS program. It is also cross-listed as an elective in the Data Science MS program, the CCIB Ph.D program and the Applied Computing MBS program. We will be using the Rutgers **Canvas** site for the course as the focal point for sharing all class material.

Attendance during class sessions is mandatory, especially since the class will run in **semi-flipped** mode with in-class activities and assessments complementing bi-weekly homework assignments.

- Beginning with week 2, the "lecture" material for the week will be made available over the prior weekend in the form of notes, assigned reading and/or videos from the resources, or short videos pre-recorded by me.
- Students are expected to complete the assigned reading/viewing **in advance of the Wednesday class sessions**.
- Each class session will consist of a sequence of activities including student-led discussions on that week's assigned readings/videos, problem-solving (theory and coding) tied to the specific learning objectives for that week, and two or three in-class assessments (quizzes, written exercises, or programming exercises).
- Bi-weekly homework (approximately 5-6 times during the semester) will be assigned for further practice and applications of class material.
- There will be a proctored, online midterm exam (around Week 7) and an in-class final exam scheduled as per the final examination calendar posted on the Registrar's site.
- Class discussions will be facilitated through the Canvas site for the course; participation in the discussions is mandatory and will be part of the grade.
- Students **must check-in** during one of the office hour sessions **at least once in two weeks**.

Tentative Schedule of Topics

Please note that this schedule is **tentative** and subject to change. The weeks refer to two consecutive class sessions and not to calendar weeks, e.g. Week 1 below includes Wednesday (Sep 6) and Monday (Sep 11).

Week	**Topics**
1	Python warmup; Summations; Arrays; Asymptotic Notation
2	Linked lists; Stacks & Queues
3	Recursion; Binary Trees; Mergesort
4	Divide and Conquer algorithms
5	Master Theorem; Quicksort
6	Random Quicksort; Hashing
7	Heaps
	Spring Break
8	Binary Search Trees; Skip Lists
9	Graphs: Breadth-First Search
10	Graphs: Depth-First Search
11	Dijkstra's Shortest-Path Algorithm
12	Minimum Spanning Trees
13	Recursion; Dynamic Programming
14	Sequence Alignment

Course Grade

The overall grade will be apportioned as follows:

- **15%** for the online midterm exam.
- **20%** for bi-weekly homework assignments (after dropping the worst score)
- **20%** for the in-class final exam.
- **45%** for the in-class assessments.

Except in the most extenuating circumstances, there will be **no makeup** opportunities for in-class assessments and the final exam. If there are medical reasons for requiring special accommodation, e.g., extra time on the in-class assessments, please obtain the relevant documents from the Division of Student Affairs and submit them to me.

Letter grade rubric

The rubric below is **not rigid**: I often use my discretion to curve the grade based on student performance:

- an A grade for overall points above 85%
- a B+ grade in the range 75 – 85%
- a B grade in the range 65 – 75%
- a C+ grade in the range 60 – 65%
- a C grade in the range 50 – 60%

Anything below 50% is considered unsatisfactory (either a D or F grade).

Academic Integrity

Student collaboration and online discussions on Canvas are certainly encouraged. However, unless noted otherwise, any required submission that will be graded (quizzes, exams) and any other normative assessment **must be completed individually without any external assistance** (that specifically means not using someone outside class to do your work, or obtaining solutions from the internet, a book or someone else in class). ”Copying” from someone else especially without citation or allowing your work to be copied by others constitutes cheating, as does plagiarism from sources including books and the web (this includes **all forms** of AI Large Language Models like ChatGPT or Bard).

Any violations of academic integrity will be dealt with immediately and severely as per the Rutgers Academic Integrity Policy and Student Code of Conduct. Please read and understand the policy carefully!