CPSC 250: Data Structure and Algorithms

Adewale Sekoni

Fall, 2019

E-mail: sekoni@roanoke.edu

Office Hours: MTWTh 12:00-1:00pm, or appointment | Class Hours: MWF 10:50-11:50pm

Office: Trexler 365B | Class Room: Trexler 363

Lab Room: Trexler 363 | Lab Hours: T 3-6pm

Course Description

In this course we will study data structures and algorithms that operate on these data structures. We will learn how to prove the correctness and efficiency of various algorithms. C++ will be our main programming language.

Required Materials

- Introduction to Algorithms, 3rd Edition, by Cormen, Leiserson, Rivest and Stein, McGraw Hill.

Prerequisites/Corequisites

CPSC 170, or permission of the instructor. Familiarity with Unix is assumed.

Course Objectives

Successful students will be able to:

1. design, implement, and test algorithms in the C++,
2. analyze the efficiency of various data structures and algorithms,
3. informally prove the correctness and efficiency of various data structures and algorithms, and
4. evaluate the practical implications of different implementations of data structures and algorithms.

Course Structure

We will meet in class for 3 hours during the week, and there will be a 3 hour laboratory period. The concepts studied in class will be complemented by several programming and laboratory assignments. In class, we will focus on theory, while the lab will focus on implementation in C++. There will be a midterm and a final exam during the semester.

In case of scheduling conflicts, make-up tests will be available by **pre-arrangement only**. Make-ups will aslo be available in case of documented medical emergencies.

Besides the exams, there will be pop quizzes in class, some homework assignments, short programming projects, and a co-curricular requirement.

This course expects you to spend at least 12 hours of work each week inside and outside of class.

Homework: On all assignments, your name must be written clearly as it appears on Inquire. Your homework must be neat and legible, you will lose points for submitting rough work.

Pop Quiz: There will be at most one pop quiz every week.

Programming projects: There will be several short programming projects assigned during the semester. Programs will be graded on correctness, style, and documentation. All programs are to be turned in electronically; instructions for submission will be given in the assignment handout. **No late work (programs, homeworks, quizzes, etc.) will be accepted**.

Co-curricular Requirement: The Mathematics, Computer Science and Physics department offers a series of discussions that appeal to a broad range of interests related to these fields of study. These co-curricular sessions will engage the community to think about ongoing research, novel applications and other issues that face these disciplines. Each student is required to attend at least three of these sessions, and turn in a short paper describing the contents of the session, and his/her critical reflections about the topic and content. **These papers are due in class within a week of the session**. A paper submitted beyond a week from the event being discussed in the paper will NOT be accepted. The MCSP Conversation Series website has the schedule of talks in the series.

Grading Policy

The final grade will be computed based on the grades in the quizzes, tests, the final exam, home works and programming projects according to the following weights:

- **3%**: Co-curricular **7%**: Quizzes **25%**: Programming Assignments
- **25%**: Homework **20%**: Midterm **20%**: Final exam

The final course grade will be calculated as follows:

- **> 92%**: A **90-92%**: A- **86-89%**: B+ **83-85%**: B **80-82%**: B- **76-79%**: C+
- **73-75%**: C **70-72%**: C- **66-69%**: D+ **63-65%**: D **60-62%**: D- **< 60%**: F

Course Policies

During Class

If you use an electronic device such as a tablet or a laptop for note-taking or to read the textbook, the content that is open on the screen should be strictly restricted to documents and pages of relevance to the class. For example, you should not have any social media websites open in your browser window, even if it is in a tab that is not currently in focus.
I encourage you to take hand written notes as you may be allowed use them during pop quizzes.
Phones are prohibited as they are rarely useful for anything in the course. Eating and drinking are allowed in class but please refrain from it affecting the course. Try not to eat your lunch in class as the classes are typically active.

Attendance Policy

Regular attendance in class is highly recommended. Regardless of attendance, students are responsible for all material covered or assigned in class.

Policies on Incomplete Grades and Late Assignments

Late assignments will be accepted for no penalty if a valid excuse is communicated to the instructor before the deadline. Otherwise, **you will receive no credit.**

Academic Integrity and Honesty

Students are expected to adhere to the Academic Integrity policies of Roanoke College. All work submitted for a grade is to be strictly the work of the student unless otherwise specified by the instructor. The policies as outlined in the Academic Integrity handbook will be enforced in the course.

Graded programs are subject to the Roanoke College Academic Integrity policies. Copying a program or a portion of a program (even a single line) or reading another person's program to obtain ideas for solving a problem is plagiarism. Other examples of integrity violation include writing code for someone else, using code written by someone else, telling someone else how to solve a problem or having someone tell you how to solve a problem (and using his/her method). These cases apply to any work that is handed in for a grade under the instructor's assumption that the work is your own. Unless specified otherwise by the instructor, discussion among students should be limited to general discussion of concepts and language details, not specific aspects of a solution to the assigned problem

Schedule and weekly learning goals

The schedule is tentative and subject to change. The learning goals below should be viewed as the key concepts you should grasp after each week, and also as a study guide before each exam.

Week 1 Mathematical Background

- Logs, Laws of Indices, Summations
- Correctness and Complexity of Algorithms
- Insertion Sort, Maximum Subarray Sum
- Lab: Vector

Week 2 Divide and Conquer

- Recursive algorithms.
- Recurrence relations, Recursive Definitions of Functions.
- Mergesort
- Lab: Experimental verification of time complexity

Week 3 Brute Force and Backtracking

- Brute Force, Divide and Conquer
- Generating Permutations and Subsets with Backtracking
- *n*-queens Problem
- Lab: *n*-queens Problem

Week 4 Linked lists, Stacks, and Queues

- Doubly Linked Lists
- Implementing Stacks and Queues with Lists
- Applications of Stacks: Matching Parenthesis, Evaluating Infix
- Lab: Evaluating Infix Using Stacks

Week 5 Heaps

- Heaps
- Heapsort
- Priority Queues
- Lab: Heap and Heap sort
- Exam Review?

Midterm: 9:50-10:50, Friday, October 11

Week 6 Fall Break

Week 7 Randomized Algorithms

- Expected Runtimes
- Random Variables, Expectations
- Random Permutations
- Lab: Loot boxes

Week 8 Quicksort, Medians and Order Statistics

- Quicksort Correctness and Complexity
- Quicksort Correctness and Complexity
- Medians and Order Statistics
- Lab: Quicksort with Function Objects

Week 9 Hash Table

- Direct-address tables
- Hash functions
- Open addressing
- Lab: Hash Table

Week 10 Binary Search Trees

- Insertion and Deletion
- Red-Black Tree Properties
- Red-Black Tree Rotation

- Lab: Red-Black Tree (Interface)

Week 11 Binary Search Trees

- Red-Black Tree Insertions
- Red-Black Trees Deletion
- Lab: Red-Black Tree (Implementation)

Week 12-13 Dynamic Programming

- Optimal Substructure
- Rod Cutting
- Longest Increasing Subsequence
- Lab: Rod cutting

Final: 8:30-11:30, Tuesday, December, 10