

Advanced Data Structures and Algorithms, Spring 2021

(SENG 2000)

Meets Synchronously Online MWF: 3:00 – 3:50 PM

Course Syllabus

1. Instructor

Jonathan A. Saddler, Ph. D

Teaching Assistant Professor of Computer Science

Department of Computer Science and Engineering, East Carolina University

Email: saddlerj20@ecu.edu and jsaddlerecu@gmail.com

Office Hours: 9:00 – 10:00 AM Tuesdays, 4:00 – 5:00 PM Wednesdays

11:00 AM – 12:00 PM Fridays

Locations for office hours will be announced in-class.

2. Course Catalog Description

Solving computational problems using maps, hash tables, skip lists, and graph data structures. Design and analysis of algorithms using recursion, dynamic programming, and branch and bound.

3. Course Web Page/Lectures

Canvas: <https://ecu.instructure.com/courses/61840>

Teams:

<https://teams.microsoft.com/l/channel/19%3abf705708b64648cab961126fdb238c6f%40thread.tacv2/General?groupId=c34fcd4-41d1-4bbc-8d98-8ea4b8e05e2f&tenantId=17143cbb-385c-4c45-a36a-c65b72e3eae8>

4. Office Hours

I will be available for office hours at the following times. To participate, navigate to our Team channel and click the “Join” link made available during the hour.

Participate whenever you need an extra boost related to course material.

OFFICE HOURS	MONDAY	TUESDAY	WEDNES.	THURSDAY	FRIDAY
		9:00 – 10:00 AM			
					11:00AM – 12:00 PM
			4:00 – 5:00 PM		

Our online Circumstances:

This course will be taught completely online, however, I will be near ECU main campus throughout the duration of this course. My office is in room 117C of the Science and Technology Building. My preferred methods of contact are as follows, in order of preference from left-to-right:

1. By Email	2. By Teams Chat	3. In Person
saddlerj20@ecu.edu is my official email address. Expect important class announcements from this address weekly. I should respond to your request within 24 hours	Our channel link is above: Use this to follow and find lecture material that we are covering in this course. I will try to upload to Canvas each week backups of our recorded sessions.	My office is 117C Sci-Tech. ECU is allowing undergraduate students to return to campus this semester. When I am available in my office, you may stop by. Please bring a mask with you to wear. I will also.

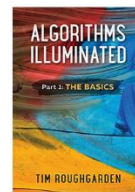
5. Required Textbooks:

You should have 4 books on hand to be fully prepared to complete this course.

[1] Roughgarden, Tim. *Algorithms Illuminated*.

Part 1: The Basics.

Soundlikeyourself Publishing, 2017, ISBN:
978-0999282908.

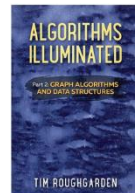


Algorithms Illuminated: Part 1: The Basics
Roughgarden, Tim
☆☆☆☆ 0 ratings by Goodreads
ISBN 10: 0999282905 / ISBN 13: 9780999282908
Published by Soundlikeyourself Publishing

[2] Roughgarden, Tim. *Algorithms Illuminated*.

Part 2: Graph Algorithms and Data Structures.

Soundlikeyourself Publishing, 2018, ISBN:
978-0999282922

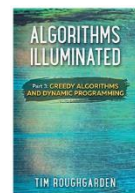


Algorithms Illuminated (Part 2): Graph Algorithms and Data Structures (Paperback or Softback)
Roughgarden, Tim
ISBN 10: 0999282921 / ISBN 13: 9780999282922
Published by Soundlikeyourself Publishing, LLC 8/5/2018, 2018

[3] Roughgarden, Tim. *Algorithms Illuminated*.

Part 3: Greedy Algorithms and Dynamic Programming.

Soundlikeyourself Publishing, 2019, ISBN:
978-0999282946.

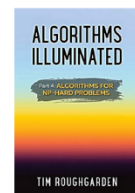


Algorithms Illuminated (Part 3): Greedy Algorithms and Dynamic Programming
Tim Roughgarden
ISBN 10: 0999282948 / ISBN 13: 9780999282946
Published by Soundlikeyourself Publishing, LLC 2/19

[4] Roughgarden, Tim. *Algorithms Illuminated*.

Part 4: Algorithms for NP-Hard Problems.

Soundlikeyourself Publishing, 2020, ISBN:
978-0999282960.



Algorithms Illuminated (Part 4): Algorithms for NP-Hard Problems
Roughgarden, Tim
ISBN 10: 0999282964 / ISBN 13: 9780999282960
Published by Soundlikeyourself Publishing, LLC 2/20

5.1 Referenced Textbooks:

[1] Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L., and Stein, Clifford. *Introduction to Algorithms*, Third. The MIT Press, 2009, ISBN: 978-0262033848 (make sure to use the 3rd Edition)

6. Course Learning Objectives

By the end of this course, students should know how to:

- Apply knowledge of computing and mathematics to analyze algorithms.
- Apply knowledge of computing and mathematics to design algorithms using recursion, dynamic programming, and branch and bound paradigms.
- Solve computational problems using lists, two dimensional arrays, maps, hash tables, skip lists, and graph data structures.
- Apply important software development paradigms to deploying codebases that rely on shared code across classes.

7. Course Topic Outline

- Sorting Algorithms and Searching Algorithms using Arrays
- Introduction to Programming Data Structures in C++
 - Variables, expressions and statements, Functions
- Algorithmic Runtime Analysis and the Master Method
- Special Data Structures in C++, including...
 - Lookup Tables (Dictionaries and Maps) and Skip Lists
 - Graphs and Trees
- Graph Search and Tree Algorithms and Path Algorithms
- If Time: “Memo”ization in Dynamic Programming/Branch-and-Bound Algorithms
- If Time: Greedy Algorithms

8. Course Assignments

Two types of assignments are to be submitted by students in this course.

“Pseudocode and Problem-solving” assignments will require that you understand the algorithm topic of the week well enough to devise solutions that utilize them to devise improvements to these algorithms. The assignments require an effort to take what was learned about algorithms in class to a further next step.

“Complete-a-program” assignments will ask you to utilize data structures that you have learned in class to implement either a full requirements specification or the remaining half of a partial implementation of one. The language that will be used to turn in assignments for this course will be **by default Python, however** a programming language that supports abstractions and object-oriented paradigms suitable to the students in the course will be determined via a poll at the start. As part of the course lecture material, students will be given a brief opportunity to learn how

to design classes and perform operations on data using C++, as electives that exist beyond this course in our curriculum require a smidge of understanding in memory management this language can teach readily to students via course content. “Complete-a-program” assignments go a step beyond “Problem-solving” assignments by writing computer programs that mimic the pseudocode using C++.

Exams are a third component of this course that will involve the recollection of specific definitions of terms defined among what you learned about algorithms. Students will demonstrate your knowledge of the structure of an algorithm, as well as the abstractions present in an algorithm, what data structures are well-known to support each algorithm, and report on strengths and weaknesses of these algorithms in various scenarios. There are three exams to be given in this course.

9. Grading

* Note: The heaviest weighted item in this course are the “complete-a-program” assignments.

Component	Relative Weight in Final Grade
Participation	8%
Pseudocode and Problem-Solving Assignments at least 3 Pseudocode and at most 2 Problem solving assignments	20%
Complete-a-Program Assignments* at least 4 complete-a-program assignments with increasing difficulty	32% potential breakdown: 2%, 6%, 12%, 12%
Exam 1	12.5%
Exam 2	12.5%
Final Exam (Wednesday, Nov. 17th, 2020)	15%

10. Grading Scale

Cutoffs for grades are listed below. Each value to the right is the lowest score that may be used to earn the letter grade on the left.

A	93%
A-	90%
B+	87%
B	83%
C+	76%
C	72%
C-	68%
D+	64%
D	60%
D-	56%

11. Participation

More to be covered in course lecture:

Participation is very important in this course. Each week from session 3 until the final day of classes, we will structure attendance gradation in tiers. All participants in the course up until the add-drop date start out in participation “Tier 8”. Missing a certain amount of class after a cutoff of dates will lower your participation grade.

Missing enough participation points will keep you from earning that well-famed “A”. Please inform me of all intended absences at least 5 hours in advance of the lecture.

12. Academic Integrity Policy for SENG 2000

“Copied code” is a bad thing and is a more self-explanatory term for “plagiarism.” When I see copied code, I will grade the assignment as usual, however, you will receive a notification for plagiarism. According to ECU Faculty Manual Part 6 on Academic Integrity Violations (below), I must pursue a specific series of actions such as special follow-up meetings to protect the integrity of all assignments submitted to our courses.

This is not a fun thing to do for either one of us. The meeting according to guidelines must happen on a strict time schedule which could interrupt your regular course schedule, and the punishment we levy could be a 0% on the assignment, a 0% on multiple assignments, or in harsher cases a failing grade for the course. Citing code that you find posted elsewhere or in someone else’s assignment will not suffice as an excuse to copy code, and such code will still be considered plagiarism if submitted.

Academic honesty is critical for success in this course. Here is a section of the faculty manual that discusses how we treat and locate plagiarism:

ECU Faculty Manual, section on Academic Integrity: <http://www.ecu.edu/cs-acad/fsonline/customcf/currentfacultymanual/part6section2.pdf>

“Plagiarism basically is submitting the work of others as your own. You may work together on homework (and are encouraged to), but you must submit your own work.”

I must add an additional statement from the department
statement on academic integrity below:

“Violations may include copying homework solutions from the posted solutions with no evidence of independent thought and submitting the work of other students as your own.”

In this course we must make an exception to this. If there are “starter solutions” that you find are posted on our course Canvas and autograder sites, you may freely copy this code to your own solution and publish it with your solution. In these situations, it is especially important to be careful about doing your own work, and not copying still others’ solutions, as files that contain starter code may still contain “copied code” inside, around, and calling starter code procedures.

Disabilities

East Carolina University seeks to comply fully with the Americans with Disabilities Act (ADA). Students requesting accommodations based on a disability must be registered with the Department for Disability Support Services located in Slay 138. ((252) 737-1016 (Voice/TTY))

Inclement Weather

In the event of a weather emergency, information about ECU can be obtained through the following sources:

ECU emergency notices	http://www.ecu.edu/alert
ECU emergency information hotline	(252) 328-0062