INFO 6205 Program Structure and Algorithms

Course Syllabus

Course Information

Professor: Nik Bear Brown
Email: ni.brown@neu.edu
Office: Zoom Only
Office hours:
Only through Zoom

Note: I am also a master's student at Northeastern. Do not send e-mail to my student e-mail brown.ni@husky.neu.edu I almost never read that e-mail.

All classes will be held via in class and through Zoom

Course website: Canvas

All office hours held through Zoom. DO NOT come to the office. Office hours are ONLY through zoom.

Course Prerequisites

A knowledge of python or some other programming language. All course examples will be in python.

Course Description

Presents data structures and related algorithms, beginning with a brief review of dynamic memory allocation. Discusses the fundamental data structures in detail, including the abstract representation of algorithms, and implementation methods. Focuses on understanding the application of the abstract data structure and the circumstances that affect implementation decisions. Covers lists, stacks, queues, trees, hash tables, and graphs. Covers recursion and searching and sorting algorithms in detail. Emphasizes data abstraction and encapsulation in code design. Explores external storage structures, time permitting.

This is an introductory course in algorithms. The focus of this course is on learning algorithm design techniques for solving computational problems.

This course teaches techniques for the design and analysis of efficient algorithms, emphasizing methods useful in practice. Topics covered include sorting, search trees, heaps, hashing, divide-and-conquer, dynamic programming, amortized analysis, graph algorithms, shortest paths, network flow, randomized algorithms, data structures, approximation algorithms, and graph search.

We will cover theory as well as implement, simulate, and visualize algorithms using python. There will be weekly homework assignments, two in-class midterm exams, a final exam, and a research project.

Learning Objectives

- Get exposed to a range of computational problems that arise in very diverse applications
- Learn how to formulate problems precisely from somewhat informal descriptions
- Learn new algorithmic design techniques used to solve the computational problems
- Learn proof techniques critical for reasoning about the correctness of algorithms
- Learn analysis techniques critical to determine the efficiency of algorithms
- Learn how to transform algorithms to programs
- Learn to implement, simulate, and visualize algorithms
- Apply algorithms to real world problems

Course Schedule

This course consists of homework assignments (every 10-15 days), 3 in class exams, quizzes and a research project. The course very closely follows the book *Algorithm Design* by Jon Kleinberg and Eva Tardos and *Python Algorithms Mastering Basic Algorithms in the Python Language* by Magnus Lie Hetland.

The topics are:

TOPIC	SLIDES	READINGS	DEMOS
Introduction (*What are algorithms?*)	Syllabus	Preface · ToC	
Stable Matching (*Gale-Shapley*)	1up · 4up	Chapter 1	Gale-Shapley
Algorithm Analysis (*big-Oh notation*)	1up · 4up	Chapter 2	
Graphs (*graph search*)	1up · 4up	Chapter 3	
Greedy Algorithms I (*basic techniques*)	1up · 4up	Chapter 4	interval scheduling · interval partitioning

Greedy Algorithms II (*shortest paths and MSTs*)	1up · 4up	Chapter 4	Dijkstra · red-blue · Prim · Kruskal · Borůvka · Edmonds
Divide and Conquer I (*sorting and selection*)	1up · 4up	Chapter 5	merging · inversions · quickselect
Divide and Conquer II (*integer and polynomial multiplication*)	1up · 4up	Chapter 5	
Dynamic Programming I (*basic techniques*)	1up · 4up	Chapter 6	
Dynamic Programming II (*sequence alignment, Bellman-Ford*)	1up · 4up	Chapter 6	
Network Flow I (*maximum flow theory*)	1up · 4up	Chapter 7	Ford-Fulkerson · pathological
Network Flow II (*maximum flow applications*)	1up · 4up	Chapter 7	
Network Flow III (*assignment problem*)	1up · 4up	Chapter 7	
Intractability I (*polynomial-time reductions*)	1up · 4up	Chapter 8	
Intractability II (*P, NP, and NP-complete*)	1up · 4up	Chapter 8	
Intractability III (*coping with intractability*)	1up · 4up	Section 10.2, 11.8	
PSPACE (*PSPACE complexity class*)	1up · 4up	Chapter 9	
Limits of Tractability (*extending limits of tractability*)	1up · 4up	Chapter 10	

Approximation Algorithms (*approximation algorithms*)	1up · 4up	Chapter 11	list scheduling
Local Search (*Metropolis, Hopfield nets*)	1up · 4up	Chapter 12	
Probability		Supplemental Material	
Randomized Algorithms (*randomized algorithms*)	1up · 4up	Chapter 13	
Data Structures I (*amortized analysis*)	1up · 4up	Supplemental Material	
Data Structures II (*binary and binomial heaps*)	1up · 4up	Supplemental Material	binary heap · heapify
Data Structures III (*Fibonacci heaps*)	1up · 4up	Supplemental Material	
Data Structures IV (*union-find*)	1up · 4up	Section 5.1.4 (*Dasgupta et al.*)	
Linear Programming I (*simplex algorithm*)	1up · 4up	Supplemental Material	
Linear Programming II (*linear programming duality*)	1up · 4up	Supplemental Material	
Linear Programming III (*ellipsoid algorithm*)	1up · 4up	Supplemental Material	

Course GitHub

The course GitHub (for all lectures, assignments and projects):

https://github.com/nikbearbrown/YouTube/tree/main/I2A_Algorithms

nikbearbrown YouTube channel

Over the course of the semester I'll be making and putting additional data science and machine learning related video's on my YouTube channel.

https://github.com/nikbearbrown/INFO_6205_Program_Structure_and_Algorithms

The purpose of these videos is to put additional advanced content as well as supplemental content to provide additional coverage of the material in the course. Suggestions for topics for additional videos are always welcome.

Teaching assistants

The Teaching assistants are:

TBA

Programming questions should first go to the TA's. If they can't answer them then the TA's will forward the questions to the Professor.

Learning Assessment

Achievement of learning outcomes will be assessed and graded through:

- Quizzes
- Completion of assignments
- Completion of term projects

Reaching out for help

A student can always reach out for help to the Professor, Nik Bear Brown ni.brown@neu.edu. In an online course, it's important that a student reaches out early should he/she run into any issues.

Grading Policies

A point system is used. Everything that you are expected to turn in has points. Points can range from 1 point to 1000 points Assignments get a 10% deduction for each day they are late rounded up. Exams cannot be made up unless arraignments are made before the exam.

I expect to use the following grading scale at the end of the semester. You should not expect a curve to be applied; but I reserve the right to use one.

Score	Grade

93 – 100	A
90 – 92	A-
88 – 89	B+
83 – 87	B
80 – 82	B-
78 – 79	C+
73 – 77	C
70 – 72	C-
60 – 69	D
<60	F

Typically grades will end up roughly 25% A, 25% A-, 25% B+, 20% B , 5% less than B but that depends on students' performance.

Canvas

You will submit your assignments via Canvas ***and*** Github. Click the title of assignment (Canvas -> assignment -> <Title of Assignment>), to go to the submission page. You will know your score on an assignment, project or test via Canvas. Canvas only represents only the raw scores. Not normalized or curved grades. A jupyter notebook file ALONG with either a .DOC or .PDF rendering of that jupyter notebook file must be submitted with each assignment.

Your name MUST be part of your submission, for example Sanchez_Rick_Assignment_1.zip

Multiple files must be zipped. No .RAR, .bz, .7z or other extensions.

Assignment file names MUST start with students last name then first name OR the groups name and include the class number and assignment number.

Assignment MUST estimate the percentage of code written by the student and that which came from external sources.

Assignment MUST specify a license at the bottom of each notebook turned in.

All code must adhere to a style guide and state which guide was used.

Due dates

Due dates for assignments at midnight on due date of the assignment.

Five percent (i.e. 5%) is deducted for each day an assignment is late. Solutions will be posted the following Monday. Assignments will receive NO CREDIT if submitted after the solutions are posted. Any extensions MUST be granted via e-mail and with a specific new due date.

Course Materials

The main texts (100% of the course)

Python Algorithms
Mastering Basic Algorithms in the Python Language
Magnus Lie Hetland
Paperback
2014
Free at https://link.springer.com/book/10.1007/978-1-4842-0055-1

Algorithm Design
Jon Kleinberg and Eva Tardos
Paperback
2013 (Any edition/version of this book is fine)

https://www.amazon.com/Algorithm-Design-Kleinberg-Jon/dp/9332518645/

Participation Policy

Participation in discussions is an important aspect on the class. It is important that both students and instructional staff help foster an environment in which students feel safe asking questions, posing their opinions, and sharing their work for critique. If at any time you feel this environment is being threatened—by other students, the TA, or the professor—speak up and make your concerns heard. If you feel uncomfortable broaching this topic with the professor, you should feel free to voice your concerns to the Dean's office.

Collaboration Policies

Students are strongly encouraged to collaborate through discussing strategies for completing assignments, talking about the readings before class, and studying for the exams. However, all work that you turn in to me with your name on it must be in your own words or coded in your own style. Directly copied code or text from any other source MUST be cited. In any case, you must write up your solutions, in your own words. Furthermore, if you did collaborate on any problem, you must clearly list all of the collaborators in your submission. Handing in the same work for more than one course without explicit permission is forbidden.

Feel free to discuss general strategies, but any written work or code should be your own, in your own words/style. If you have collaborated on ideas leading up to the final solution, give each other credit on what you turn in, clearly labeling who contributed what ideas. Individuals should be able to explain the function of every aspect of group-produced work. Not understanding what plagiarism is does not constitute an excuse for committing it. You should familiarize yourself with the University's policies on academic dishonesty at the beginning of the semester. If you have any doubts whatsoever about whether you are breaking the rules – ask!

Any submitted work violating the collaboration policies WILL BE GIVEN A ZERO even if "by mistake." Multiple mistakes *will be sent to OSCCR for disciplinary review.*

To reiterate: **plagiarism and cheating are strictly forbidden. No excuses, no exceptions.** *All incidents of plagiarism and cheating will be sent to OSCCR for disciplinary review.*

Assignment Late Policy

Assignments are due by 11:59pm on the due date marked on the schedule. It is your responsibility to determine whether or not it is worth spending the extra time on an assignment vs. turning in incomplete work for partial credit without penalty. Any exceptions to this policy (e.g. long-term illness or family emergencies) must be approved by the professor.

Assignments will receive NO CREDIT if submitted after the solutions are posted. Any extensions MUST be granted via e-mail and with a specific new due date.

Only ONE extension will be granted per semester.

Student Resources

Special Accommodations/ADA: In accordance with the Americans with Disabilities Act (ADA 1990), Northeastern University seeks to provide equal access to its programs, services, and activities. If you will need accommodations in this class, please contact the Disability Resource Center (www.northeastern.edu/drc/) *as soon as possible* to make appropriate arrangements, and please provide the course instructors with any necessary documentation. The University requires that you provide documentation of your disabilities to the DRC so that they may identify what accommodations are required, and arrange with the instructor to provide those on your behalf, as needed.

Academic Integrity: All students must adhere to the university's Academic Integrity Policy, which can be found on the website of the Office of Student Conduct and Conflict Resolution (OSCCR), at http://www.northeastern.edu/osccr/academicintegrity/index.html. Please be particularly aware of the policy regarding plagiarism. As you probably know, plagiarism involves *representing anyone else's words or ideas as your own*. It doesn't matter where you got these ideas—from a book, on the web, from a fellow-student, from your mother. It doesn't matter whether you quote the source directly or paraphrase it; if you are not the originator of the words or ideas, *you must state clearly and specifically where they came from*. Please consult an instructor if you have any confusion or concerns when preparing any of the assignments so that together. You can also consult the guide "Avoiding Plagiarism" on the NU Library Website at http://www.lib.neu.edu/online_research/help/avoiding_plagiarism/. If an academic integrity concern arises, one of the instructors will speak with you about it; if the discussion does not resolve the concern, we will refer the matter to OSCCR.

Writing Center: The Northeastern University Writing Center, housed in the Department of English within the College of Social Sciences and Humanities, is open to any member of the

Northeastern community and exists to help any level writer, from any academic discipline, become a better writer. You can book face-to-face, online, or same day appointments in two locations: 412 Holmes Hall and 136 Snell Library (behind Argo Tea). For more information or to book an appointment, please visit http://www.northeastern.edu/writingcenter/.