Syllabus

This is a single, concatenated file, suitable for printing or saving as a PDF for offline viewing. Please note that some animations or images may not work.

Description

This module is also available as a concatenated page, suitable for printing or saving as a PDF for offline viewing.

MET CS 566
Analysis of Algorithms

This course teaches theoretical backgrounds for design and analyzing algorithms, as well as practical implementation methods. The course starts with a review of principles of algorithm analysis and includes divide and conquer, dynamic programming, greedy programming, matrix operations, and extend them to advance topics of linear programming. Students should be familiar with basic data structures and basic Python programming. Weekly course assignments include both theoretical analysis and practical algorithmic implementation in python.

Prerequisite: MET CS 521 (Information Structures with Python) and MET CS 526 (Data Structures and Algorithms), or instructor’s consent

Technical Notes

The table of contents expands and contracts (+/- sign) and may conceal some pages. To avoid missing content pages, you are advised to use the next/previous page icons in the top right corner of the learning modules.

This course requires you to access files such as word documents, PDFs, and/or media files. These files may open in your browser or be downloaded as files, depending on the settings of your browser.

Learning Objectives

By successfully completing this course, you will be able to:

- Implement algorithm with the theoretical backgrounds of computer science analysis and design, as well as practical implementation methods.
- Understand the concepts of asymptotic notation in the analysis of algorithms and its usage in comparing algorithm performance.
- Understand the concepts of divide and conquer algorithms and its usage in algorithm design.
- Understand the concepts of hashing, binary search trees, graph algorithms, and dynamic programming.
- Describe advance analysis of algorithm topics like NP-Completeness and NP-Hard problems.

Instructor

Joshua Enxing
Lecturer
Computer Science Department
Metropolitan College
Boston University

Email: jenxing@bu.edu

Joshua Enxing has been involved with teaching in the MET CS department since 2012 when he was an undergraduate at Boston University. Currently, in the Metropolitan College he teaches CS677 and CS566, and facilitates CS546, while teaching various math and computer science courses at other Boston-area universities. He has worked in many different positions in the technical field, among them are software developer, data scientist, and statistical programmer. During his time at Tufts University, he served as Vice President and then President of the Tufts Chapter of SIAM (Society of Industrial and Applied Mathematics). In conjunction with colleagues at Tufts University, he was part of NSF-funded research dealing with novel methods for diffuse optical tomography image reconstruction. Current areas of research include nonparametric statistics, machine learning, and data science.

Course Developer: Kia Teymourian

Dr. Teymourian holds a PhD from Freie Universität Berlin as well as a MS and BS from Berlin University of Technology (TU-Berlin). His computer science expertise lies in data stream processing and complex event processing, big data programming, semantic technologies, and knowledge representation, as well as web technologies and natural language processing. He has made important contributions to multiple large and international research projects, including several funded by the European Commission, the German Federal Ministry of Education and Research (BMBF), and the DARPA Pliny Project at Rice University. He is a senior member of Institute of Electrical and Electronics Engineers (IEEE), and a member of the Association for Computing Machinery (ACM).

Additional information can be found on Dr. Teymourian's Academic Website.

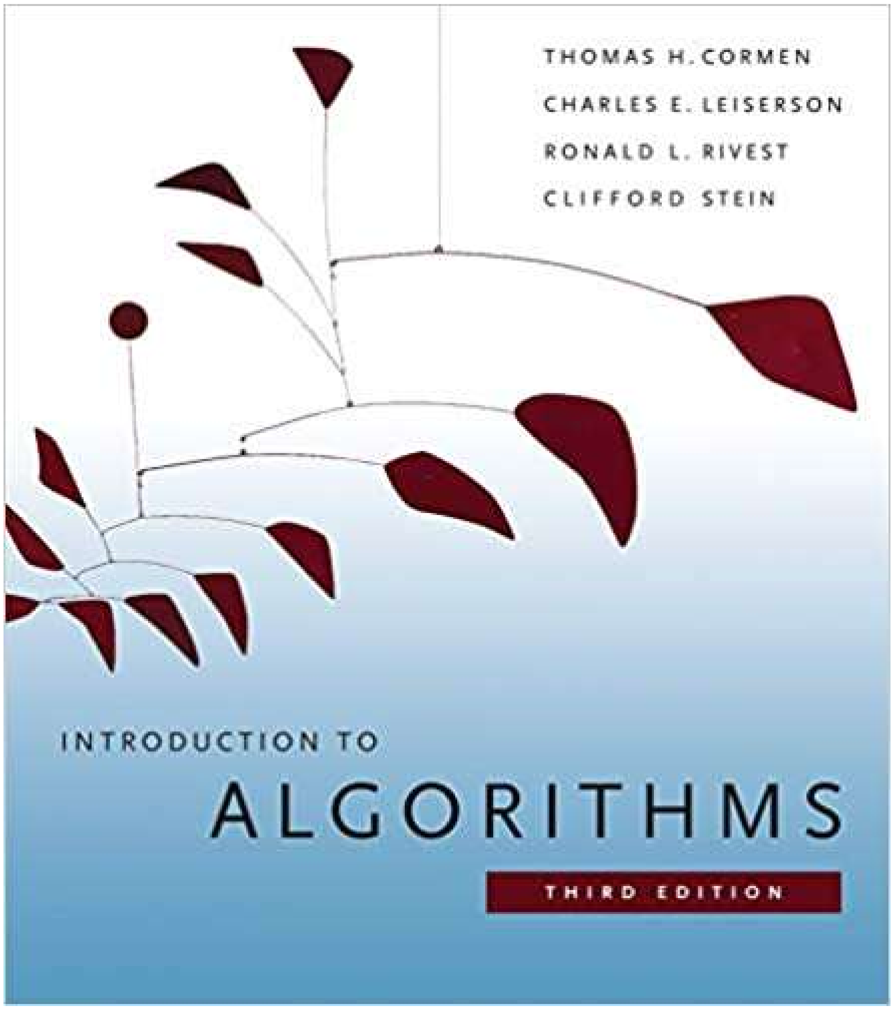
Materials

Required Book

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms*. 3rd ed.
The MIT Press.

ISBN: 978-0262033848.

This book can be purchased from Barnes and Noble at Boston University. An e-book is available from the MIT Press.



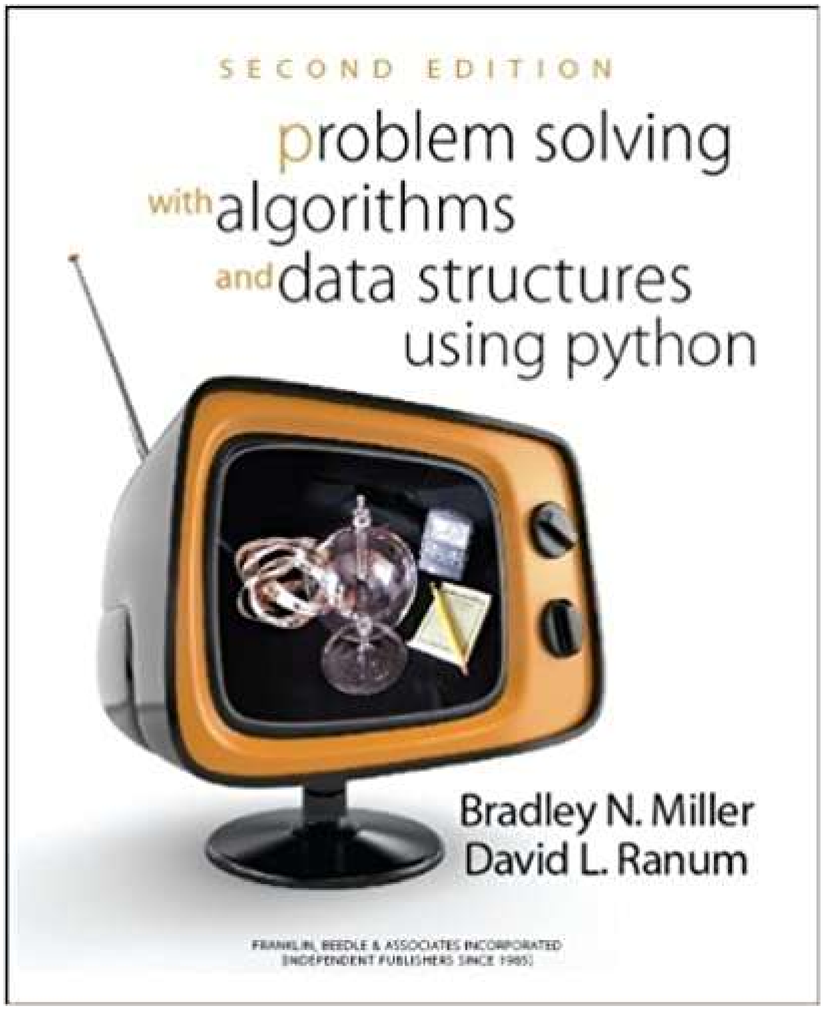
Note: We refer to the book as CLRS book in the course.

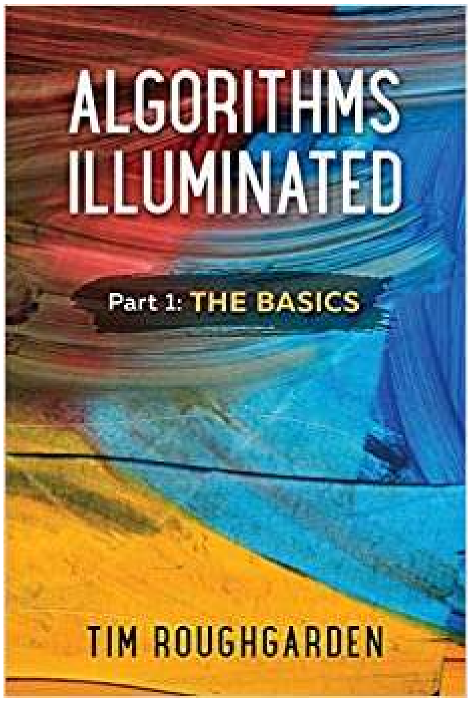
Recommended Books

There will be no reading assignments from the recommended books.

Miller, B., & Ranum, D. (2011). *Problem Solving with Algorithms and Data Structures Using Python*. 2nd ed.
Franklin, Beedle & Associates.

ISBN: 978-1590282571.



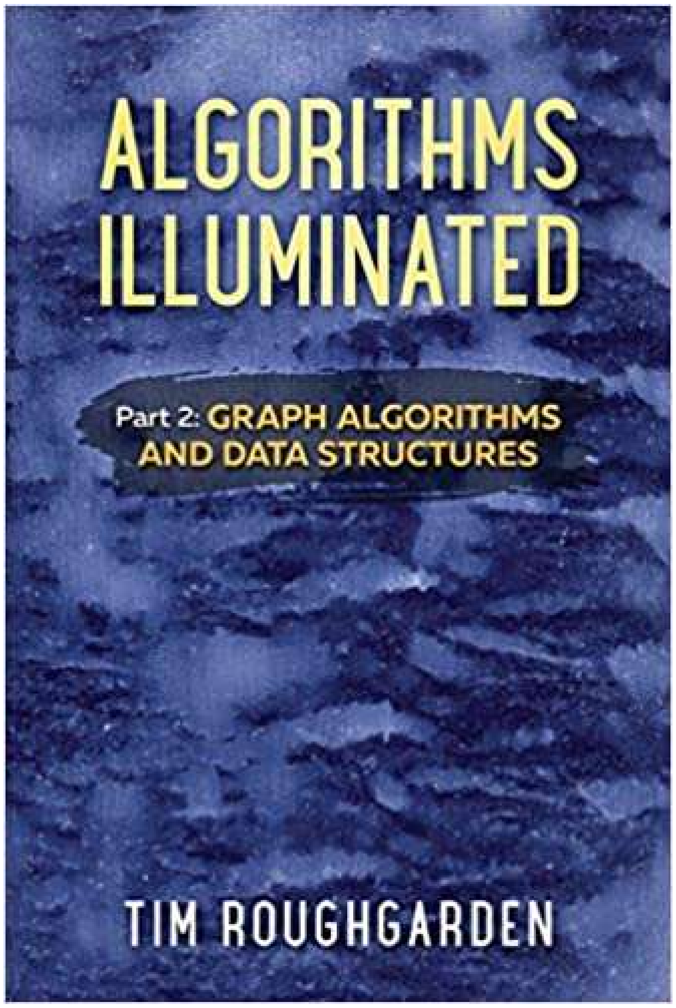


Roughgarden, T. (2017). *Algorithms Illuminated (Part 1): The Basics*.
Soundlikeyourself Publishing.

ISBN: 978-0999282908.

Roughgarden, T. (2018). *Algorithms Illuminated (Part 2): Graph Algorithms and Data Structures*.
Soundlikeyourself Publishing.

ISBN: 978-0999282922.



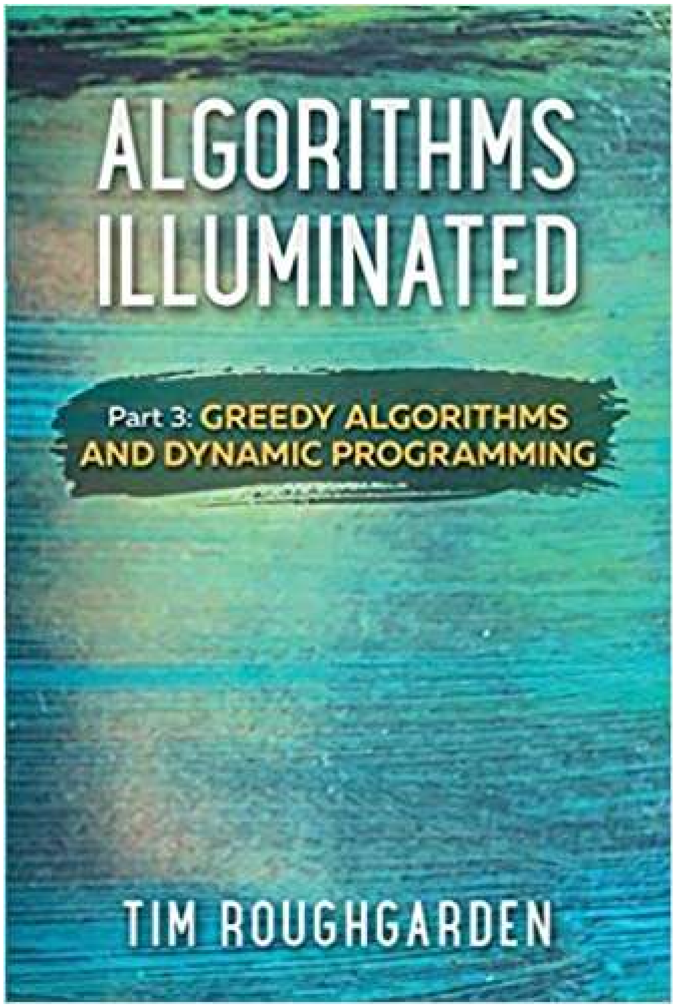
ALGORITHMS ILLUMINATED

Part 2: GRAPH ALGORITHMS AND DATA STRUCTURES

TIM ROUGHGARDEN

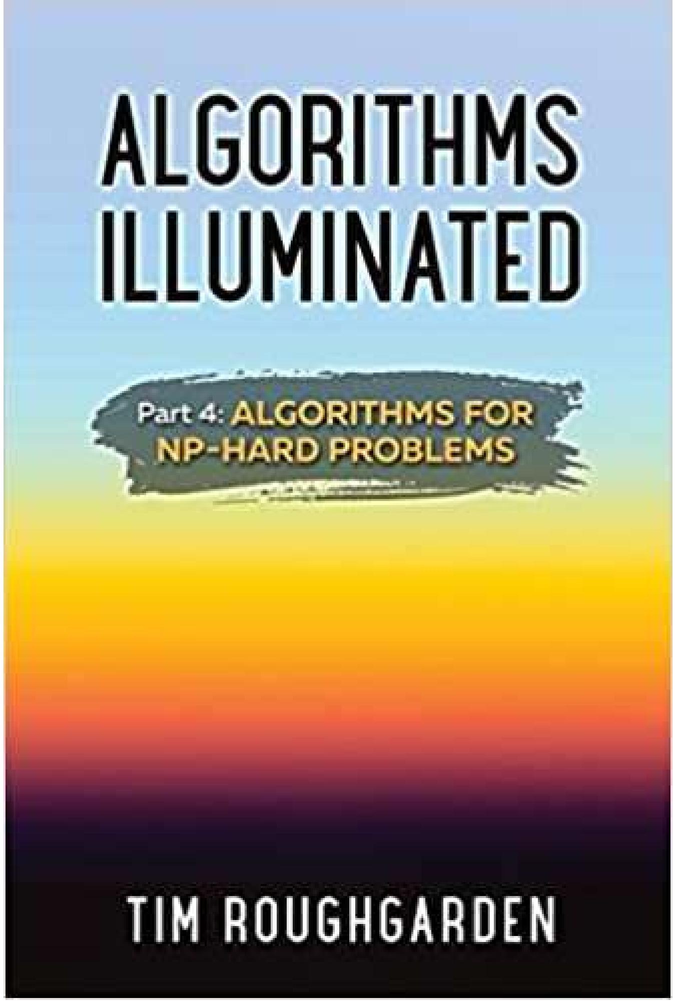
Roughgarden, T. (2019). *Algorithms Illuminated (Part 3): Greedy Algorithms and Dynamic Programming*. Soundlikeyourself Publishing.

ISBN: 978-0999282946.



Roughgarden, T. (2020). *Algorithms Illuminated (Part 4): Algorithms for NP-Hard Problems*. Soundlikeyourself Publishing.

ISBN: 978-0999282960.



Boston University Library Information

Boston University has created a set of videos to help orient you to the online resources at your disposal. An introduction to the series is below:

met_ode_library_14_sp1_00_intro video cannot be displayed here

All of the videos in the series are available on the Online Library Resources page, which is also accessible from the Campus Bookmarks section of your Online Campus Dashboard. Please feel free to make use of them.

As Boston University students, you have full access to the BU Library. From any computer, you can gain access to anything at the library that is electronically formatted. To connect to the library, use the link http://www.bu.edu/library. You may use the library's content whether you are connected through your online course or not, by confirming your status as a BU community member using your Kerberos password.

Once in the library system, you can use the links under “Resources” and “Collections” to find databases, eJournals, and eBooks, as well as search the library by subject. Some other useful links follow:

Go to Collections to access eBooks and eJournals directly.

If you have questions about library resources, go to Ask a Librarian to email the library or use the live-chat feature.

To locate course eReserves, go to Reserves.

Please note that you are not to post attachments of the required or other readings in the water cooler or other areas of the course, as it is an infringement on copyright laws and department policy. All students have access to the library system and will need to develop research skills that include how to find articles through library systems and databases.

Free Tutoring Service

Free online tutoring with Smarthinking is available to BU online students for the duration of their courses. The tutors do not rewrite assignments, but instead teach students how to improve their skills in the following areas: writing, math, sciences, business, ESL, and Word/Excel/PowerPoint.

You can log in directly to Smarthinking from Online Campus by using the link in the left-hand navigation menu of your course.

YouTube

Please Note

Smarthinking may be used only for current Boston University online courses and career services. Use of this service for purposes other than current coursework or career services may result in deactivation of your Smarthinking account.

Study Guide

Module 1 Study Guide and Deliverables

Theme:	Review of Principles of Algorithm Analysis
Topics:	• What is an Algorithm? • Growth of Functions • Asymptotic Notation • Big Theta, Big O and Big Omega Notation • Insertion sort • Asymptotic Costs of Programs • Big O of Python Code Snippet
Readings:	Lecture material
Assignments:	Assignment 1 due **Monday, January 24 at 6:00 AM ET.**

- Submit at “Assignments” on the left-hand course menu.

Live Classroom: Live lecture will include 60 minutes lectures and 30 minutes Q&A at the end of each lecture.

- Recorded session
- Thursday, January 20 at 8:00 – 9:00 PM ET
- Live office hours with a facilitator: TBD

Module 2 Study Guide and Deliverables

Theme: Divide and Conquer, Sorting

Topics:

- Divide and Conquer – Merge sort
- Divide and Conquer – Strassen’s algorithm
- Recurrences
- Recursion-tree method
- Heaps and Heap sort

Readings: Lecture material

Assignments: Assignment 2 due **Monday, January 31 at 6:00 AM ET**

- Submit at “Assignments” on the left-hand course menu.

Live Classroom: Live lecture will include 60 minutes lectures and 30 minutes Q&A at the end of each lecture.

- Monday, January 24 at 7:00 – 8:00 PM ET
- Thursday, January 27 at 8:00 – 9:00 PM ET
- Live office hours with a facilitator: TBD

Module 3 Study Guide and Deliverables

Theme: Heapsort, Hashing and Searching

Topics:

- Hash Tables
- hashing with chaining
- Amortized Analysis
- Binary Search Trees
- Insertion and Deletion in Trees

Readings: Lecture material

Assignments:

- Assignment 3 due **Monday, February 7 at 6:00 AM ET** (through "Assignments")
- The term project guideline will be published at the end of Module 3.

Live Classrooms: Live lecture will include 60 minutes lectures and 30 minutes Q&A at the end of each lecture.

- Monday, January 31 at 7:00 – 8:00 PM ET
- Thursday, February 3 at 8:00 – 9:00 PM ET
- Live office hours with a facilitator: TBD

Module 4 Study Guide and Deliverables

Theme: Graphs

Topics:

- Breadth-first search (BFS)
- Depth-first search (DFS), topological sorting
- Single-source shortest paths problem
- Dijkstra
- Bllman-Ford

Readings: Lecture material

Assignments:

- Assignment 4 due **Monday, February 14 at 6:00 AM ET** (through "Assignments")
- Term Project proposal due Thursday, February 10 at 6:00 AM ET (through "Assignments")
- Begin work on Midterm assignment. You have two weeks to complete, which is due by the end of Module 5.

Live Classroom: Live lecture will include 60 minutes lectures and 30 minutes Q&A at the end of each lecture.

- Monday, February 7 at 7:00 – 8:00 PM ET
- Thursday, February 10 at 8:00 – 9:00 PM ET
- Live office hours with a facilitator: TBD

Module 5 Study Guide and Deliverables

Theme: Dynamic Programming

Topics:

- Elements of dynamic programming
- Fibonacci
- Shortest Paths
- The Principles of Dynamic Programming
- Text justification, blackjack

Readings: Lecture material

Assignments:

- Assignment 5 due **Monday, February 21 at 6:00 AM ET** (through "Assignments")
- Midterm Assignment due Thursday, February 17 at 6:00 AM ET (through "Assignments")

Live Classroom: Live lecture will include 60 minutes lectures and 30 minutes Q&A at the end of each lecture.

- Monday, February 14 at 7:00 – 8:00 PM ET
- Thursday, February 17 at 8:00 – 9:00 PM ET
- Live office hours with a facilitator: TBD

Module 6 Study Guide and Deliverables

Theme: BST, Greedy Algorithms and Computational Complexity

Topics:

- Parenthesization, edit distance, knapsack (Dynamic Programing)
- Recursive Activity Selector (Greedy)
- Computational complexity
- P and NP, NP-Completeness, NP-Hard Problems

Readings: Lecture material

Assignments:

- Term Project Presentation due Thursday, February 24 at 6:00 AM ET
    - Share video presentation at "Media Gallery" on the left-hand course menu.
    - **How to record a video and share at the "Media Gallery" section?** Check out the direction to use Kaltura to capture and post or submit video.
    - Submit presentation slides and programming files at "Assignments" on the left-hand course menu.

Live Classroom: Live lecture will include 60 minutes lectures and 30 minutes Q&A at the end of each lecture.

- **Sunday, February 20** at 7:00 – 8:00 PM ET
- Office hour: Thursday, February 24 at 8:00 – 9:00 PM ET
- Live office hours with a facilitator: TBD

Final Exam Details

The Final Exam is a proctored exam available from **Friday, February 25 at 6:00 AM ET to Monday, February 28 at 11:59 PM ET**.

The Computer Science department requires that all final exams be administered using an online proctoring service called Examity that you will access via your course in Blackboard. In order to take the exam, you are required to have a working webcam and computer that meets Examity's system requirements. A detailed list of those requirements can be found on the How to Schedule page ("Proctored Final Exam Information" module at the course home page). Additional information regarding your proctored exam will be forthcoming from the Assessment Administrator. You will be responsible for scheduling your own appointment within the defined exam window.

The Final Exam is accessible only during the final exam period. You can access it from the "Assessments" section of the course. Your proctor will enter the password to start the exam.

Final Exam Duration: two hours.

It is an **open-book/open-note exam**: any print and electronic materials are allowed, including electronic versions of textbooks, lecture module PDFs, and internet resources.

Grading Information

Grading Structure and Distribution

The grade for the course is determined by the following, including both theoretical algorithmic analysis as well as practical implementation in python:

Overall Grading Percentages

Five Assignments	40

Midterm Assignment	20
Term Presentation	10
Proctored Final Exam	30

Graded Items:

- **Assignments**: From Module 1 to Module 5, there is one assignment at the end of each module. There are a total of five assignments.
  - You can access and submit the assignment at the “Assignments” section on the left-hand course menu.
- **Midterm Assignment**: A midterm assignment is similar to other module assignments but includes more advanced tasks. It includes questions related to the module 1 to the end of module 3. Students will have two weeks to complete the midterm assignment.
  - Access and submit the midterm assignment at the “Assignments” section on the left-hand course menu.
- **Term Project Presentation**: Students learn one topic out of a list of topics, and prepare a presentation video and/or implementation of the algorithm.
  - Term project guidelines will be published at the end of Module 3.
  - In the final project presentation, students will present a topic and record a video between 8 to 12 minutes
    - Share the term project presentation video at the “Media Gallery” section.
    - **How to record a video and share at the “Media Gallery” section?** Check out the direction to use Kaltura to capture and post or submit video.
- **Proctored Final Exam:** There will be a proctored Final Exam in this course using a proctor service called Examity. Detailed instructions regarding your proctored exam will be forthcoming from the Assessment Administrator. You will be responsible for scheduling your own appointment.
  - Access and take the Final Exam at the “Assessments” section on the left-hand course menu.

Class Policies

1. Assignment Completion & Late Work: We recognize that emergencies occur in professional and personal lives. If one occurs that prevents your completion of homework by a deadline, please share the plan with your facilitator or instructor. This must be done in advance of the deadline (unless the emergency makes this impossible, of course), and should be accompanied by particulars that back it up. Additional documentation may be requested. Late submissions without reasons will result in grade deduction. **Late homework can be accepted up to 48 hours after the due date. 10% penalty will be applied after 24 hours and 20% after 48 hours, and after that we will not accept any late submissions.**

2. Academic Conduct Code: Cheating and plagiarism will not be tolerated in any Metropolitan College course. They will result in no credit for the assignment or examination and may lead to disciplinary actions.

Ungraded Items:

- **Ungraded Discussion Board:** There are ungraded discussion forums throughout the course, access at the “Class Discussion” section on the left-hand course menu. You are encouraged to ask course related questions, share your knowledge, and learn from your peers. Discussions forums are provided for your benefit. Some discussion forums involve the instructional staff; others are among students.