Boston University Metropolitan College

Analysis of Algorithms

Spring 2021 – MET CS-566
On Campus

Ed Orsini
edorsini@bu.edu
Office hours: by appointment

Meets Tuesdays at 6:00PM – 8:45PM EST (Starting on 1/26)

Course Description

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.

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.

Recommended Books

The following book is required.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). ***Introduction to Algorithms*****. 3rd ed.**
Publisher: The MIT Press.
ISBN: 978-0262033848
This book can be purchased from Barnes and Noble at Boston University.

The following books are recommended, but are not required.

Miller, B., & Ranum, D. (2011). ***Problem Solving with Algorithms and Data Structures Using Python*****. 2nd ed.**
Publisher: Franklin, Beedle & Associates.
ISBN: 978-1590282571

Roughgarden, T. (2017). ***Algorithms Illuminated (Part 1): The Basics*****.**
Publisher: Soundlikeyourself Publishing.
ISBN: 978-0999282908

Roughgarden, T. (2018). ***Algorithms Illuminated (Part 2): Graph Algorithms and Data Structures*****.**
Publisher: Soundlikeyourself Publishing.
ISBN: 978-0999282922

Roughgarden, T. (2019). ***Algorithms Illuminated (Part 3): Greedy Algorithms and Dynamic Programming*****.**
Publisher: Soundlikeyourself Publishing.
ISBN: 978-0999282946

Roughgarden, T. (2020). ***Algorithms Illuminated (Part 4): Algorithms for NP-Hard Problems*****.**
Publisher: Soundlikeyourself Publishing.
ISBN: 978-0999282960

Courseware

Blackboard site: https://onlinecampus.bu.edu/

Course Prerequisites
Students should have a solid background in object-oriented programming. The following classes are required/recommended:

- MET CS 521 (Information Structures with Python) and MET CS 526 (Data Structures and Algorithms), or instructor's consent

Fall 2020 COVID-19 Policies

Compliance: All students returning to campus will be required, through a digital agreement, to commit to a set of Health Commitments and Expectations including face coverings, symptom attestation, testing, contact tracing, quarantine, and isolation. The agreement makes clear that compliance is a condition of being a member of our on-campus community.

You have a critical role to play in minimizing transmission of COVID-19 within the University community, so the University is requiring that you make your own health and safety commitments. Additionally, if you will be attending this class in person, you will be asked to show your Healthway badge on your mobile device to the instructor in the classroom prior to starting class, and wear your face mask over your mouth and nose at all times. If you do not comply with these rules you will be asked to leave the classroom. If you refuse to leave the class, the instructor will inform the class that they will not proceed with instruction until you leave the room. If you still refuse to leave the room, the instructor will dismiss the class and will contact the academic Dean's office for follow up.

Boston University is committed to offering the best learning environment for you, but to succeed, we need your help. We all must be responsible and respectful. If you do not want to follow these guidelines, you must participate in class remotely, so that you do not put your classmates or others at undue risk. We are counting on all members of our community to be courteous and collegial, whether they are with classmates and colleagues on campus, in the classroom, or engaging with us remotely, as we work together this fall semester.

Class Policies

1. **Attendance & Absences** – I will not be taking attendance. It is the student's responsibility to keep up with the material covered in class.
2. **Assignment Completion & Late Work** – Late work will not be accepted. We recognize that emergencies occur in professional and personal lives. If one occurs that prevents your completion of homework by a deadline, please make your instructor aware as soon as possible. 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: we want to be fair to everyone in this process, including the vast majority of you who sacrifice so much to submit your homework on time in this demanding schedule.
3. **Academic Conduct Code** – Please use the following wording, or an equivalent, in your syllabus: "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. Please take the time to review the Student Academic Conduct Code:

http://www.bu.edu/met/metropolitan_college_people/student/resources/conduct/code.html.

NOTE: [This should not be understood as a discouragement for discussing the material or your particular approach to a problem with other students in the class. On the contrary – you should share your thoughts, questions and solutions. Naturally, if you choose to work in a group, you will be expected to come up with more than one and highly original solutions rather than the same mistakes.]

Grading Criteria

The grading percentages for the course is determined by the following:

Assignments	40%
Midterm Assignment	20%
Term Presentation	10%
Final Exam	30%

Translation between letter grades and percentages:

A (Excellent)	95-100
A- (Excellent; minor improvement evident)	90-94.99
B+ (Very good)	87-89.99
B (Good)	83–86.99
B- (Good mostly some significant improvements needed)	80-82.99
C+ (Satisfactory; some significant improvements needed)	77-79.99
C (Satisfactory; significant improvements needed)	73–76.99
C- (Satisfactory; significant improvements required)	70-72.99
D Many improvements required	65
Fail	0

Graded Items

- **Assignments**: There are a total of 5 assignments. From Module 1 to Module 5, there is one assignment at the end of each module.
- **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.

- **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 to the class.

Class Meetings, Lectures & Assignments

Lectures, Readings, and Assignments subject to change, and will be announced in class as applicable within a reasonable time frame.

Date	Topic	Readings Due	Assignments Due
January 26	Review of Principles of Algorithm Analysis What is an Algorithm? Growth of Functions Asymptotic Notation	Module 1 online content	
February 2	Big Theta, Big O and Big Omega Notation Insertion sort Asymptotic Costs of Programs Big O of Python Code Snippet	Module 1 online content	
February 9	Divide and Conquer, Sorting	Module 2 online content	Assignment 1 due Tuesday, 2/9, at 6:00 AM ET

	Divide and Conquer – Merge sort Divide and Conquer – Strassen's algorithm Recurrences		
February 16	Recursion-tree method Heaps and Heap sort	Module 2 online content	
February 23	Heapsort, Hashing and Searching Hash Tables hashing with chaining Amortized Analysis	Module 3 online content	Assignment 2 due Tuesday, 2/23, at 6:00 AM ET
March 2	Binary Search Trees Insertion and Deletion in Trees	Module 3 online content	
March 9	Breadth-first search (BFS) Depth-first search (DFS), topological sorting Single-source shortest paths problem	Module 4 online content	Assignment 3 due Tuesday, 3/9, at 6:00 AM ET
March 16	Dijkstra Bellman-Ford	Module 4 online content	
March 23	Dynamic Programming	Module 5 online content	Assignment 4 due Tuesday, 3/23, at 6:00 AM ET

	Elements of dynamic programming Fibonacci Shortest Paths		
March 30	The Principles of Dynamic Programming Text justification, blackjack	Module 5 online content	Midterm Assignment due Tuesday, 3/30, at 6:00 AM ET
April 6	BST, Greedy Algorithms and Computational Complexity Parenthesizing, edit distance, knapsack (Dynamic Programing) Recursive Activity Selector (Greedy)	Module 6 online content	
April 13	Computational complexity P and NP, NP-Completeness, NP-Hard Problems	Module 6 online content	Assignment 5 due Tuesday, 4/13, at 6:00 AM ET
April 20 (no class)			
April 27	Presentations		
May 4	Final Exam		