Course Information and Policy

EC504 Fall 2019 Tuesday, Thursday
11:00am -12:45pm LSE-B03

Administrative Staff

Instructors

Rich Brower: brower@bu.edu, 353-6052, PRB 581

Office hours: By appointment via Email request and/or Skype calls.

Assistants

Mert Toslali: toslali@bu.edu

Yahia Bakour: yehia234@bu.edu

Office hours: At end of Class and ???

Course Content

Algorithms and their associated Data structures are widely employed to improve computational efficiency by orders of magnitude. We will begin the course with a quick review of: (i) analysis techniques for algorithms and data structures (ii) C or C++ syntax and language features, and (iii) simple data structures. We will then proceed to examine more complicated data structures, including balanced search trees and priority queues, with applications to databases and query processing. Thereafter we will examine graph-based structures, including traversals, spanning trees, shortest paths, and flows, within the context of networks. We will conclude with a choice of advanced topics, such as NP completeness, approximation algorithms, game solvers, Quantum Computing, Machine Learning concepts together with applications. Throughout the course we will focus on rigorous analysis and realistic applications.

Prerequisite

The prerequisite for this course is EC327 - Introduction to Software Engineering or an equivalent undergraduate course that utilizes a modern languages.

Textbooks

[CLRS] Cormen, Leiserson, Rivest, and Stein, *Introduction to Algorithms* (Third Edition), MIT press, 2009: Reading assignments will be in this text. This is the most complete reference for data structures and algorithms currently in use, and it is found on the bookshelves of many professional engineers. **(required)**

Optional reference text

Mark Allen Weiss, *Data Structures & Algorithm Analysis in Java* (Second Edition) Addison- Wesley, 1998: This is a version of an alternate text used for the undergraduate version of this class some semesters. It is not nearly as comprehensive as our textbook [CLRS], but it is somewhat easier to understand and provides C code, not just pseudocode.

On-line references:

Cplusplus explains the C++ language from its basics up to the newest features introduced by C++11. Chapters have a practical orientation, with example programs in all sections to start practicing what is being explained right away.

http://www.cplusplus.com/

In class labs

As part of this course, you will access the CCS in the classroom (LSE-B03) for the programming exercise. The class project is **alg504**. Notes will be provided to using your CCS accounts. The development can be done on personal laptop using your favorite editor and IDE. However the final version must be place in the your CCS directory and compile with the standard Makefiles.

Online Documents

Course GitHUB. https://github.com/brower/EC504. You are responsible for checking this regularly. It will contain handouts, homeworks, and related material.

Grades

All grades will be curved according to the class average. Thus, it is only your *relative* score (compared to the rest of the class) that matters, rather than your numerical score. For a course at this level, I expect to center the average at a B/B+, but the final grade will depend on my assessment of the class as a whole

Composition

Raw scores will be computed based on the following approximate weights:

Homeworks (25%) – including coding exercises
Class participation (5%)
Project (20%)
Midterm exam (25 %) – Oct24 in class
Final exam (25 %) – Dec ??, ??PM- ??PM

Homeworks

Homeworks for written exercise are due at the **end** of class. There will be a 10% penalty per day for late homework, but no homeworks will be accepted after the start of the next class. Penalties may be removed only for valid excuses with written, dated documentation.

Coding exercise will be graded by compiling the source code on CCS with you Makefile and running it against a series of standard input files. The deadline will be compared with the unix time stamp of your files. The format input and output will be fixed so the basic grading can be automated. You will be able to try your source code multiple time against standard input and compare output files distributed with the prototype main routine. Comparison with this standard will verify it correctness on this file but the full test will include additional input files to make sure nothing is hardwired to get the correct solutions. The grade will be given on the last version before the deadline.

Lecture Notes

You will each be responsible for taking notes for one lecture during the semester to demonstrate complete understanding of some specific course material. You will submit your lecture notes (and choose which one you wish to write) as a pdf file which will be posted on the web site for your fellow students.

Lecture notes must be submitted within *one week* of their lecture, after which point your class mates will rank the notes based on how useful, accurate, and complete they found them. Your grade will be based on this ranking.

Project

You will complete one final project involving an implementation of some interesting algorithm or data structure. There will be a Final Project handout soon with details and deadlines for this project.

Extra credit

Extra credit opportunities will be given out throughout the semester, typically in class participation and occasionally for graphics output from coding exercises presented the class.

Collaboration

You may use any textbooks or web sources when completing your homework, and/or **one** (no more) human collaborator (from class) per homework, subject to the following strictly enforced conditions:

1. You must clearly acknowledge all your sources (including your collaborator) on the top of your homework. You must write all answers in your own words.

2. The may discuss the programming exercise with any one in the class BUT the actually C or C++ code must be yours. Do not copy line by line from other students or for web examples?

3. You must be able to fully explain your written answers and the function of your codes upon demand (and I will demand it!)

You may not use *any human resource* outside of class (including web-based help services, outside tutors,etc.) in doing your homeworks or project. Obviously, you may not collaborate with anyone on exams.

Failure to meet any of the above conditions could constitute plagiarism and will be considered cheating in this class. If you are not sure whether something is permitted by the course policy, ASK !