COURSE INFORMATION AND POLICY

Tuesday, Thursday 11:00 to 12:45pm, CDS 263

Staff

Instructor

Rich Brower: brower@bu.edu,

Office hours: By appointment.

Assistant

Yiqin Zhang: evezhang@bu.edu

Office hours: To be determined

Course Content

Data structures are widely employed to store data in an efficient manner, and can 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 syntax and language features (all C++ compilers compile C and you may use C++ syntax sugar if you wish), and (iii) simple data structures. We will then proceed to examine more complicated data structure containers, 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 possible advanced topics, such as NP completeness, approximation algorithms and simple numerical methods, even a bit of machine leaning and quantum computing! Throughout the course we will focus on rigorous analysis, benchmarking performance w.r.t scaling and timing on CCS.

Prerequisite

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

Textbooks

[CLRS] Cormen, Leiserson, Rivest, and Stein, *Introduction to Algorithms* (Third Edition), MIT press, 2009: 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].

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/

Labs

The virtual lab is Massacchusetts Green High Performance Computing Center (MGHPCC) http://www.bu.edu/tech/support/research/rcs/mghpcc/ with account on the Sheared Compute Clusters (SCC) http://www.bu.edu/tech/services/research/computation/scc/ This is modern HPC center. As is best practice you can use your laptop and the environment that fits your style and then the final code is run and benchmarked on a *real computer*.

Online

Collaboration and communication is done with the industrial tools of Slack and GitHub. No silly institutional tools that are not useful in the real world. GitHub will contain handouts, homeworks, and related material. May set preference to notified on any changes

Grades

All grades will be curved according to the class average. Thus, it is your *relative* score (compared to the rest of the class) that really matters, rather than your objective 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 (40%)
Project and in Class (20%)
Midterm exam (20%)
Final exam (20%)

Homeworks

Homeworks are due at 11:59PM for Software on CCS
for written execise on GradeScope.

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.

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 collaborators) on the top of your homework.
2. You must write all answers for the written exerisea in your own words and for the software exercise the C code must be written by you. Credit will depend on the clarity and simplicity of the written exercises, the code and the out put table and figures. (More is not a virtue unless for extra credit you are inspired to go beyond the exercise with extra functionality. This can be presented as class participation.)