EC 504: Advanced Data Structures and Algorithms
Fall, 2024

Lecture Time

TTh 11:00am-12:45pm, CDS 262

Staff Information

Instructor: Prof. Brian Kulis
Office: PHO 441
Email: bkulis@bu.edu (please use EC504 in the subject line when emailing)
Office hours: Tuesday 3-4pm, PHO 441

GST: Wentao Shangguan (wentaos@bu.edu)
Office hours: Monday 6-7:30pm, PHO 442

Course Description

This course will focus on data structures and algorithm design. As a graduate course in this area, the course will focus more on analysis and theoretical aspects rather than implementation details. Students are expected to be able to design and analyze new algorithms for problems based on the techniques discussed in class. Homeworks will be a mix of pen-and-paper exercises and implementation exercises. There will also be a larger project component done in groups.

Prerequisites

For undergraduates, EC330 (Applied Algorithms and Data Structures) is required. It is assumed that all students have taken a course in basic data structures and algorithms and are familiar with the following: sorting, asymptotic analysis, recurrence relations, basic data structures such as linked lists, stacks and queues, hash tables, binary search trees, and heaps. Strong discrete math skills are crucial for the algorithmic techniques discussed in class, particularly induction proofs. Some background in probability theory is also necessary for randomized algorithms, which may be covered depending on progress.

Topics

We will cover several topics in advanced data structures and algorithm design. In particular, we will focus on (tentatively):

1. Introductory Material and Review
    a. Asymptotic Analysis
    b. Review of Basic Data Structures (Binary Search Trees, Heaps)
2. Advanced Data Structures
    a. B-Trees
    b. Van Emde Boas Trees
    c. Fibonacci Heaps

        d. Disjoint Set Data Structures
3. Algorithmic Design
    a. Divide and Conquer
    b. Greedy Algorithms
    c. Dynamic Programming
    d. Network Flow
4. Intractability
    a. NP-Completeness
    b. PSPACE-Completeness
    c. Approximation Algorithms
5. Selected Topics
    a. Randomized Algorithms and Data Structures

Webpage

Announcements, course material, readings, and an updated schedule will be posted to on piazza.

Textbooks

Required: Cormen, Leiserson, Rivest, and Stein. *Introduction to Algorithms*. MIT Press, 2009 (3rd Edition). This book is exhaustive, and covers nearly all of the topics in the course (and many more that we will not have time to cover). It is an excellent reference to own.

Extremely Useful: Kleinberg and Tardos. *Algorithm Design*. Addison-Wesley, 2005. This book will be on reserve at the Science and Engineering Library. Some of the lectures will follow the treatment given by this book, and the material on PSPACE will come from this text. I find this book to be fairly easy to follow. It focuses on algorithm design, and thus does not cover most of the data structures material.

Homework

There will be 6-8 homeworks during the course. *Homeworks are to be handed in on gradescope.* The homeworks will be a mix of pen-and-paper exercises along with implementation exercises throughout the course. For the implementation questions, you will be required to write a program for the specified task; we will provide base code. The code will be written in C/C++..

Important notes on homework:

When a question asks to design an algorithm, you need to provide a description of your solution with pseudo-code as well as a proof of correctness and an analysis of the running time. The clearest way to explain an algorithm is to use English, with some notation. Solutions that consist only of pseudo-code tend to be indecipherable by anyone except the author. These solutions also tend to have inaccuracies and are often incorrect. We reserve the right to deduct a significant number of points for solutions that consist only of pseudo-code with no explanation.

Homeworks can be done in groups (and in fact this is highly encouraged), but homework exercises need to be written up individually. Also, write down the names of any students with whom you collaborated. You must be able to fully explain your answers on demand, if necessary. You may not use resources outside of class including web-based services, etc; failure to do so may be considered plagiarism and will be taken very seriously.

Regarding regrades, for any grade you think is incorrect, write down a written explanation of the grading error and drop it off before or after class. The graders will take a look at these requests periodically. We will typically not look at regrade requests that are only arguing about the amount of partial credit.

The late policy for homeworks is that every day that the homework is late, there is a 15 percent penalty; after 3 days no credit will be given. So, up to one day late, you can receive a maximum of 85%; up to two days late, a maximum of 70%, and up to three days late, a maximum of 55%. After three days, you will receive a zero.

Project

There will be a class project for the course. This will involve a significant implementation of a data structure or algorithm discussed in the course, or a structure or algorithm not discussed in the class but which is related to the course material. There will be example projects to choose from on the course webpage; alternatively, students can design their own project. Groups of up to 4 are permitted; smaller groups are OK but may have trouble finishing the project on time. You will be required to provide a project writeup of up to 10 pages describing the project and the results. Code will also be submitted as part of the project.

The last two classes will consist of project presentations. Depending on the number of groups, presentations will be 10-15 minutes long.

Exams

There will be a midterm exam in-class (date TBD). This will be closed-book. There will be a final exam on.

Grade Breakdown

Midterm: 20%
Final: 20%
Written Homework: 20%
Software Homework: 20%
Project (including presentation): 20%