CSCE 3110: Data Structures and Algorithms - Fall 2019

Instructor: Paul Tarau, Professor - see my home page for contact info and office hours.

course directory

related code at github

Grader(s): TBD, contact via Canvas

Midterm exam: TBD, open net

Final exam: TBD, cumulative, open net

Description and Objectives:

This course is intended to emphasize the understanding of non-linear data structures and elementary graph algorithms. We will cover both theoretical analysis and experimentation. Programming assignments will focus on the design of elegant data types and implementation of practical and efficient algorithms.

Syllabus

- Intro to basic data types in Python 3
  - Dictionaries, lists/arrays, sets, stacks, queues
  - List and set comprehensions, generators

- Overview of simple recursive data types and algorithms
- Overview of Object Oriented Programming, classes and objects in Python

- Time and Space analysis
  - Asymptotic notation
  - Analysis of Sorting Algorithms
    - Insertion sort
    - Merge sort
    - Quick sort
• Overview of pointer and memory operations in C

• Implementing Python's List Arrays: Dynamic Arrays in C

• Implementing Dictionaries: Hash tables in C

• Amortized algorithm analysis

• Data types for trees

• Recursion and Recurrence relations

• Tree algorithms, including
  • Generating trees of a given size
  • Balanced Trees
  • Evaluation of Expression Trees
  • Evaluation of Boolean Formulas

• Data types for graphs, including
  • adjacency lists
  • matrices

• Using Python's Graph Processing Libraries, graph visualization

• Graph Algorithms, including
  • Depth-first search
  • Breadth-first search
  • Iterative deepening
  • Connected components
  • Topological ordering
  • Prim’s algorithm
  • Kruskal’s algorithm
  • Graph coloring
  • Boolean Circuit Evaluation
  • PageRank

**Evaluation:**

• 2 Individual Exams: 30%+30%
Recommended books and online materials:

Problem Solving with Algorithms and Data Structures using Python, by Bradley N. Miller, also here or here.

Data Structures and Algorithm Analysis in C++ by Mark A. Weiss, 4-th edition

Software, tutorials and related links:

- Recurrence relations
- Solving Recurrence Equations, 2 methods
- On Solving Recurrence Equations Symbolically
- Python 3 tutorial
- A nice DS and Algorithms course
- Another nice DS course and another one
- Functional Python 3 also, a good blog on it
- Python 3 generators and list-comprehensions
- Python 3 itertools library
- Python Graph Library

Outcomes:

- Understand time complexity of algorithms.
- Be able to solve recurrence relations.
- Understand and be able to analyze the performance of data structures for searching, including balanced trees, hash tables, and priority queues.
- Apply graphs in the context of data structures, including different representations, and analyze the usage of different data structures in the implementation of elementary graph algorithms including depth-first search, breadth-first search, topological ordering, Prim's algorithm, and Kruskal's algorithm.
- Be able to code the above-listed algorithms.