CSE 310 Data Structures and Algorithms
Syllabus

Note: detailed syllabus with deadlines and course schedule will be posted in the canvas shell before semester starts)

1. Course Information
   a. Catalog Description
      Advanced data structures and algorithms, including stacks, queues, trees (Red-Black trees, B, B+, AVL), and graphs. Searching in graphs, hashing, external sorting.

   b. Prerequisites
      MAT 243 Discrete Mathematical Structures
      CSE 240 Introduction to Programming Languages or CSE 220 Programming for Computer Engineering

2. Textbook (Required)
   Cormen, T. H., Leiserson, C.E., and Rivest, R.L., *Introduction to Algorithms,* 3rd ed., The MIT Press.

3. Course Goals
   Students who complete this course can
   - define data structures (types) such as heaps, balanced trees, hash tables.
   - explain how to use a specific data structure in modeling a given problem (e.g. I can explain how to model a dictionary using balanced trees).
   - identify, construct, and clearly define a data structure that is useful for modeling a given problem.
   - state some fundamental algorithms such as merge sort, topological sort, Kruskal's algorithm, and algorithmic techniques such as dynamic programming and greedy algorithms.
   - use a specific algorithmic technique in solving a given problem (e.g. I can write a dynamic program that solves a shortest-path problem).
   - design an algorithm to solve a given problem.
   - define the notions of worst-case/best-case/average-case running times of algorithms.
   - analyze and compare different asymptotic running times of algorithms.
   - analyze a given algorithm and determine its asymptotic running time.
   - combine fundamental data structures and algorithmic techniques in building a complete algorithmic solution to a given problem.
   - combine fundamental data structures and algorithmic techniques in building a complete algorithmic solution to a given problem.
   - create several algorithmic solutions to a given problem and choose the best one among them according to given requirements on time and space complexity.

4. Topics
- Asymptotic Notation
- Recursion, recurrence relations
- Worst-case Analysis
- Divide-and-conquer.
- Sorting and sorting lower bound.
- Median, Selection Problems
- Hash Tables
- Binary Search Trees, Red-Black Trees
- Dynamic Programming
- Longest Common Subsequence
- Greedy algorithms
- Graph Algorithms: Depth-first and breadth-first search, shortest path, topological sort
- Disjoint Sets (union-find): Introduction to Amortized Analysis.

5. Class Format/Attendance:

In-person lectures and recitations. You are free to attend any of the recitations available during the week. Recitations will be hosted by the TA and will cover a brief practice that reviews the lectures of the previous week and is expected to be turned in in-class.

The attendance is recorded in terms of the recitation completeness.

6. Grading

Assessment & Grading

Assessment	**Percentage (Tentative)**
Exams (two midterms + final)	40%
Recitation / Attendance	5%
Assignments (Written / Coding Projects)	55%
Total	**100%**

Final Grade Percentage (Tentative)

A+ >= 96%
A >= 90% & < 96%
A- >= 87% & < 90%
B+ >= 84% & < 87%
B >=80% & < 84%
B- >=77% & < 80%
C+>= 74% & < 77%
C >=70% & < 74%
D >= 60%& <70%
E < 60%

May be adjusted slightly, but no general curve-fitting.