# CS 610-103:  DATA STRUCTURES & ALGORITHMS

| **Fall 2023** | **CKB 217** | **Tue 6:00 PM - 8:50 PM** |
|---|---|---|

**Instructor**: Marzieh Eskandari     **Email:** marzieh.eskandari@njit.edu     **Webex**: https://njit.webex.com/meet/me374

| **Office:** GITC 4313 | **Office Hours:** Tue 4:30-5:30, Fri 4:30-5:30 |
|---|---|

**Prerequisites:**

1. Undergrad course on Data Structures & Algorithms (CS 505 or equivalent);
2. Discrete Math (CS 506 or CS 241 or equivalent);
3. Programming Maturity.

**Textbook:**

Michael Goodrich and Roberto Tamassia,
Algorithm Design: Foundations, Analysis, and Internet Examples, Wiley, 2002.
ISBN: 0-471-38365-1.  (Available at NJIT bookstore)
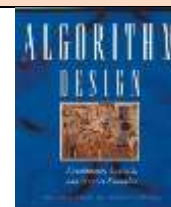
**Evaluation:**

Assignments: 20%
Attendance: 5%
Class Activity: +5% (Extra)
Midterm Exam (**Oct 31st**): 35%
Final Exam (**Dec**): 40%

**Note 1:**  NJIT Picture ID required for all exams.  All exams are closed books and closed notes.

**Course Description:**

This is a graduate-level course on data-structures and algorithms, with an emphasis on algorithm design techniques and analysis of algorithms. Topics include analysis techniques, worst-case and average-case analysis, recursion, recurrence relations, priority queues, hash tables, binary-search trees, balanced search trees (AVL trees, red-black trees), sorting algorithms; divide-and-conquer design technique and other design techniques such as greedy-method and dynamic-programming, graph algorithms.

**Course Objectives (what you are expected to get out of this course):**

1. Learn basic analysis techniques
2. Learn basic design techniques
3. Learn recurrence equations and how they are used in analysis of algorithms
4. Learn advanced data structures: Priority queues, heaps, hash tables, and search trees
5. Understand sorting algorithms and their complexities
6. Learn basic graph algorithms and their applications

**Grading:**

The grading scale (out of 100) is: 90–100: A, 80–89: B+, 70–79: B, 60–69: C+, 50 – 59: C.

| | | |
|---|---|---|
| | | |

**Schedule of Assignments & Due Dates:**

| Assignment | Subject | Date |
|---|---|---|
| Homework 1 | Week 1,2,3 | Sep 19- Sep 26 |
| Homework 2 | Week 4,5,6 | Oct 10-Oct 17 |
| **Midterm Exam** | Time complexity, Recursive Algorithms, Linked-Lists, stack, Queue, Trees, Priority Queues, Heaps, Heapsort, Sorting Algorithms, Dictionaries and Hash Tables, Balanced Search Trees | Oct 31 |
| Homework 3 | Week 8,10 | Nov 7- Nov 14 |
| Homework 4 | Week 11,12 | Nov 28- Dec 5 |
| **Final Exam** | D&C, Greedy, DP | Dec |

**Course Outline:**

| Week | Date | Topic |
|---|---|---|
| 1 | Sep 5 | Introduction, Analysis Techniques, Examples of worst-case and average-case analysis, Complexity definitions: O( ), Omega, Theta |
| 2 | Sep 12 | Recursive Algorithms, Recurrence Relations, Binary Search |
| 3 | Sep 19 | Lists, Stacks, Queues, Trees |
| 4 | Sep 26 | Priority Queues, Heaps, Heapsort |
| 5 | Oct 3 | Sorting Algorithms: Insertion-Sort, Bubble-Sort, Selection-Sort, Merge-Sort, Quicksort, Integer Sorting: Bucket-Sort, Radix Sort, ...., Lower-Bound on Sorting by Comparison |
| 6 | Oct 10 | Dictionary ADT and Hash Tables |
| 7 | Oct 17 | Balanced Search Trees: AVL and Red-Black trees<br>Graphs: Definitions, Representations and Traversals |
| 8 | Oct 24 | Divide-and-Conquer (Strassen's Matrix Multiplication, Large Integer Multiplication, Min & Max, ...), |
| **9** | **Oct 31** | **Review/Midterm Exam** |
| 10 | Nov 7 | Greedy method (Fractional Knapsack, Task Scheduling, Huffman Coding, Single-Source-Shortest-Paths (Dijkstra)) |
| 11 | Nov 14 | Greedy method (MST Algorithms: Prim, Kruskal, ...) |
| | Nov 21 | **Thanksgiving** |
| 12 | Nov 28 | Dynamic Programming (introduction, Binomial coefficients, All-Pairs-Shortest-Paths (Floyd)) |
| 13 | Dec 5 | Dynamic Programming (Chain Multiplication, Optimal Binary Tree Search, ...) |
| 14 | Dec12 | Review/Class work |