Spring 2023

# CS 610: Data Structures and Algorithms

Marzieh Eskandari

NJIT

Department of Computer Science

# CS 610-006: DATA STRUCTURES & ALGORITHMS

| Spring 2023 | KUPF211 | Tue 2:30 PM - 5:20 PM |
|---|---|---|

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

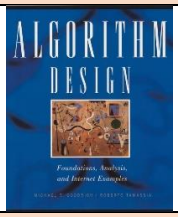
| **Office:** GITC 4313 | **Office Hours:** Tue 12:45-2:15, Fri 4:45-5:45 |
|---|---|

**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: 15%
Attendance: 5%
Midterm Exam 1 (**February 14**): 20%
Midterm Exam 2 (**March 28**): 20%
Makeup Exam (**April 25**)
Final Exam (**May**): 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, 40-49: D

**Academic Integrity:**

Academic Integrity is the cornerstone of higher education and is central to the ideals of this course and the university. Cheating is strictly prohibited and devalues the degree that you are working on. As a member of the NJIT community, it is your responsibility to protect your educational investment by knowing and following the academic code of integrity policy that is found at: http://www5.njit.edu/policies/sites/policies/files/academic-integrity-code.pdf

Please note that it is my professional obligation and responsibility to report any academic misconduct to the Dean of Students Office. Any student found in violation of the code by cheating, plagiarizing or using any online software inappropriately will result in disciplinary action. This may include a failing grade of F, and/or suspension or dismissal from the university. If you have any questions about the code of Academic Integrity, please contact the Dean of Students Office at dos@njit.edu

**Schedule of Assignments & Due Dates:**

| Assignment | Subject | Due Date |
|---|---|---|
| Homework 1 | - | Jan 31 |
| **Midterm Exam 1** | Time Complexity, Recursive Algorithms, Lists, Stacks, Queues, Trees, Priority Queues, Heaps, Heapsort | Feb 14 |
| Homework 2 | - | Mar 21 |
| **Midterm Exam 2** | Sorting, Dictionary and Hash Tables, Balanced Search Trees and Graphs | Mar 28 |
| Homework 3 | - | Apr 18 |
| **Final Exam** | Divide-and-Conquer, Greedy, Dynamic Programing | May |

**Course Outline:**

| Week | Date | Topic |
|---|---|---|
| 1 | Jan 17 | Introduction, Analysis Techniques, Examples of worst-case and average-case analysis, Complexity definitions: O( ), Omega, Theta |
| 2 | Jan 24 | Recursive Algorithms, Recurrence Relations, Binary Search |
| 3 | Jan 31 | Lists, Stacks, Queues, Trees |
| 4 | Feb 7 | Priority Queues, Heaps, Heapsort |
| 5 | Feb 14 | Review/Midterm Exam 1 |
| 6 | Feb 21 | Sorting Algorithms: Insertion-Sort, Bubble-Sort, Selection-Sort, Merge-Sort, Quicksort, Integer Sorting: Bucket-Sort, Radix Sort<br>Lower-Bound on Sorting by Comparison |
| 7 | Feb 28 | Dictionary ADT and Hash Tables |
| 8 | Mar 7 | Balanced Search Trees: AVL and Red-Black trees<br>Graphs: Definitions, Representations and Traversals |
| Spring break | | |
| 9 | Mar 21 | Divide-and-Conquer (Strassen's Matrix Multiplication, Large Integer Multiplication, Min & Max) |
| 10 | Mar 28 | Review/Midterm Exam 2 |
| 11 | Apr 4 | Greedy method (Fractional Knapsack, Task Scheduling, Huffman Coding, Single-Source-Shortest-Paths (Dijkstra)) |
| 12 | Apr 11 | Greedy method (MST Algorithms: Prim, Kruskal)<br>Dynamic Programming (introduction, Binomial coefficients) |
| 13 | Apr 18 | Dynamic Programming (All-Pairs-Shortest-Paths (Floyd), Matrix Chain Multiplication, Optimal Binary Tree Search) |
| 14 | Apr 25 | Review/Makeup Exam |