

## CS 610-1J1: Data Structures & Algorithms-Fall 2024

**Class Location:** JC 02

**Class Time:** Thu 6:00 PM - 8:50 PM

**Instructor:** Marzieh Eskandari

**Email:** [marzieh.eskandari@njit.edu](mailto:marzieh.eskandari@njit.edu)

**Zoom:** <https://njit-edu.zoom.us/j/meskandari>

**Office:** GITC 4313

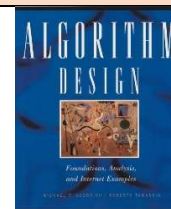
**Office Hours:** Thu 4 PM- 5:45 PM

### 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% (Extra)  
 Class Activity: +5% (Extra)  
 Midterm Exam 1 (**Oct 9th**): 25%  
 Midterm Exam 2(**Nov 6th**): 25%  
 Final Exam (**Dec**): 35%

**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

<b>Grading:</b>		
The grading scale (out of 100) is: 90–100: <b>A</b> , 1 <sup>st</sup> quarter: <b>B+</b> , 2 <sup>nd</sup> quarter: <b>B</b> , 3 <sup>rd</sup> quarter: <b>C+</b> , 4 <sup>th</sup> quarter: <b>C</b> , <60: <b>F</b> .		
<b>Academic Integrity:</b>		
<p>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: <a href="http://www5.njit.edu/policies/sites/policies/files/academic-integrity-code.pdf">http://www5.njit.edu/policies/sites/policies/files/academic-integrity-code.pdf</a></p> <p>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 <a href="mailto:dos@njit.edu">dos@njit.edu</a></p>		
<b>Schedule of Assignments &amp; Due Dates:</b>		
<b>Assignment</b>	<b>Subject</b>	<b>Date</b>
Homework 1		Sep 19- Sep 25
<b>Midterm Exam 1</b>	Time complexity, Recursive Algorithms, Linked-Lists, stack, Queue, Trees, Priority Queues, Heaps, Heapsort	Oct 10
Homework 2		Oct 24-Oct 30
<b>Midterm Exam 2</b>	Sorting Algorithms, Dictionaries and Hash Tables, Balanced Search Trees, Graphs	Nov 7
Homework 3		Nov 21- Nov 27
<b>Final Exam</b>	D&C, Greedy, DP	Dec
<b>Course Outline:</b>		
<b>Week</b>	<b>Date</b>	<b>Topic</b>
1	Sep 5	Introduction, Analysis Techniques, Complexity definitions: O, Omega, Theta
2	Sep 12	Recursive Algorithms, Recurrence Relations (Review)
3	Sep 19	Stacks, Queues, Linked-Lists, Trees (Review), Trie data Structure
4	Sep 26	Priority Queues, Heaps, Heapsort
5	Oct 3	Sorting: Insertion-Sort, Bubble-Sort, Selection-Sort, Merge-Sort, Quicksort (Review) Integer Sorting: Bucket-Sort, Radix Sort (MSD, ...), Lower-Bound on Sorting by Comparison, Shell Sort, Bitonic Sort and Counting Sort
<b>6</b>	<b>Oct 10</b>	<b>Review/Midterm Exam 1: Weeks 1,2,3,4</b>
7	Oct 17	Dictionary ADT and Hash Tables
8	Oct 24	Balanced Search Trees: AVL and Red-Black trees, Graphs: Representations and Traversals, Topological Sorting
9	Oct 31	Divide-and-Conquer: Strassen's Matrix Multiplication, Large Integer Multiplication (Karatsuba), Min & Max, FFT, VEB(if time allows)
<b>10</b>	<b>Nov 7</b>	<b>Review/Midterm Exam 2: Weeks 5,7,8</b>
11	Nov 14	Greedy method (Fractional Knapsack, Task Scheduling, Huffman Coding)
12	Nov 21	Greedy method (MST Algorithms: Prim, Kruskal, Single-Source-Shortest-Paths (Dijkstra)...) )
13	Nov 26	Dynamic Programming: introduction, Binomial coefficients, All-Pairs-Shortest-Paths (Floyd)
	<b>Nov 28</b>	<b>Thanksgiving</b>
14	Dec 5	Dynamic Programming: Chain Multiplication, Optimal Binary Search Tree Problem Solving Prep: Final Exam Readiness