New Jersey Institute of Technology
Digital Commons @ NJIT

Computer Science Syllabi | NJIT Syllabi

Spring 2024

CS 610-006: Data Structures and Algorithms

Marzieh Eskandari

Follow this and additional works at: https://digitalcommons.njit.edu/cs-syllabi

Recommended Citation
Eskandari, Marzieh, "CS 610-006: Data Structures and Algorithms" (2024). *Computer Science Syllabi*. 438.
https://digitalcommons.njit.edu/cs-syllabi/438

This Syllabus is brought to you for free and open access by the NJIT Syllabi at Digital Commons @ NJIT. It has been accepted for inclusion in Computer Science Syllabi by an authorized administrator of Digital Commons @ NJIT. For more information, please contact digitalcommons@njit.edu.

NJIT

Department of Computer Science

CS 610-006: DATA STRUCTURES & ALGORITHMS

Spring 2024	CKB 217	Thu 1:00 PM - 3:50 PM

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

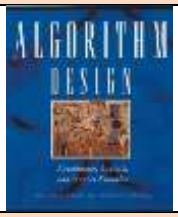
Office: GITC 4313	**Office Hours:** Thu 4:00-5:00, Fri 4:00-5:00

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%
Class Activity: +5% (Extra)
Midterm Exam (**March 7th**): 40%
Makeup Exam (**April 4th**)
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**, 1st quarter: **B+**, 2nd quarter: **B**, 3rd quarter: **C+**, 4th quarter: **C**, <50: **F**.

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	Date
Homework 1		Feb 3- Feb 10
Homework 2		Feb 24-Mar 2
Midterm Exam	Time complexity, Recursive Algorithms, Linked-Lists, stack, Queue, Trees, Priority Queues, Heaps, Heapsort, Sorting Algorithms, Dictionaries and Hash Tables, Balanced Search Trees	Mar 7
Homework 3		Mar 30- Apr 20
Final Exam	D&C, Greedy, DP	May

Course Outline:

Week	Date	Topic
1	Jan 18	Introduction, Analysis Techniques, Examples of worst-case and average-case analysis, Complexity definitions: O( ), Omega, Theta
2	Jan 25	Recursive Algorithms, Recurrence Relations
3	Feb 1	Lists, Stacks, Queues, Trees
4	Feb 8	Priority Queues, Heaps, Heapsort
5	Feb 15	Sorting Algorithms: Insertion-Sort, Bubble-Sort, Selection-Sort, Merge-Sort, Quicksort, Integer Sorting: Bucket-Sort, Radix Sort, ..., Lower-Bound on Sorting by Comparison
6	Feb 22	Dictionary ADT and Hash Tables
7	Feb 29	Balanced Search Trees: AVL and Red-Black trees Graphs: Definitions, Representations and Traversals
8	**Mar 7**	**Review/Midterm Exam**
	Mar 14	**Spring Recess**
9	Mar 21	Greedy method (Fractional Knapsack, Task Scheduling, Huffman Coding)
10	Mar 28	Greedy method (MST Algorithms: Prim, Kruskal, Single-Source-Shortest-Paths (Dijkstra)...)
11	Apr 4	Divide-and-Conquer (Mergesort, Strassen's Matrix Multiplication, ...) + **Makeup Exam**
12	Apr 11	Divide-and-Conquer (Karatsuba Integer Multiplication, Min & Max, VEB, ...) Dynamic Programming (introduction, Binomial coefficients, ...)
13	Apr 18	Dynamic Programming (All-Pairs-Shortest-Paths (Floyd), Chain Multiplication, Optimal Binary Tree Search,...)
14	Apr 25	Review+ Problem Solving Prep: Final Exam Readiness