

Fall 2024

CS 435-005: Advanced Data Structures and Algorithm Design

David Nassimi

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

Recommended Citation

Nassimi, David, "CS 435-005: Advanced Data Structures and Algorithm Design" (2024). *Computer Science Syllabi*. 505.

<https://digitalcommons.njit.edu/cs-syllabi/505>

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.

CS 435 Syllabus Sections 3,5 Fall 2024	Data Structures & Algorithms	Prof. David Nassimi (Revised 8/29/24)
	Section 3: T 4-5:20, Kup 117 R 4-5:20, GITC 1400 Recitation F 4-5, Kup 117	
	Section 5: TR 1-2:20, CULM Lect 2 Recitation F 1-2, CULM Lect 2	

Prof.: David Nassimi Zoom: https://njit-edu.zoom.us Email: nassimi@njit.edu Office: GITC 4308, Phone: 973-596-5645	TA: Genwei Zhang Email: gz6@njit.edu
Canvas: canvas.njit.edu Course materials are posted on Canvas. All assignments must be submitted on Canvas.	

Office Hours (GITC 4308): T,R 3:00-3:45, 6:00-7:00

Online Class Meeting on Zoom, in case instruction switches to online First, logon Zoom using your UCID & Password. (This is a necessary first step.) Then, input Passcode or Meeting ID.
Section 3 (Starting 4 pm): Passcode: 977 709 Meeting ID: 927 5182 6503 Link: https://njit-edu.zoom.us/j/92751826503?pwd=0RrMOme2KOYfOO67YQcoLMXOWSmdrp.1
Section 5 (Starting 1 pm): Passcode: 210 259 Meeting ID: 987 0737 5845 Link: https://njit-edu.zoom.us/j/98707375845?pwd=tH5uiaZ9ivU1bM2nsMCaY6iJmGa5q1.1

Textbook:

Authors	T.H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein
Title	Introduction to Algorithms
Edition	Third Edition, MIT Press, 2009
ISBN-13	978-026203384-8

Prerequisites:	1. CS 241 (Discrete Math) 2. CS 288 (Intensive Programming)
-----------------------	--

Evaluation:	Assignments	36%
	Midterm Exam	32%
	Final	32%

Note: 1) The average of your midterm and final exams must be at least 50.
 2) The average of all assignments must also be at least 50.

Policies:

1. Assignments must be done by you individually (no team-work).
2. Programming assignments must be in C, C++, JAVA, or Python.
3. Use of Generative AI not allowed.
4. All assignments must be submitted electronically on Canvas by the due date.
Late Penalty: 10% per day, up to 3 days. Not accepted after 3 days.

Submission of Assignments: There are two types of assignments:

- **Homework Assignments:** These are problem sets and may be either typed or hand-written neatly and scanned and uploaded in PDF format. (When a homework asks for an algorithm, it means pseudo-code on paper.)
- **Programming Assignments:** These assignments must be formally implemented and run on a computer to produce results. Submissions of these assignments must be in multiple files as follows. (Please DO NOT zip your files into one, as the TA may have problem unzipping it.) The multiple files are:
 1. **Source code:** This is the file you used to run the program, which has a suffix .cpp, .c, .java, .py, etc., depending on the programming language. The TA needs to read this file to evaluate your program, and run the program to verify that it works. (A PDF file would be useless.)
 2. Input data, if relevant, and if not too large
 3. Output produced by the program, if relevant, and if not too large
 4. Discussion and Analysis, if needed.

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

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

1. Learn basic analysis techniques
2. Learn various design techniques
3. Understand induction, recursion, and proof techniques
4. Learn recurrence equations and how they are used in analysis of algorithms
5. Understand sorting algorithms, their complexities, and how they compare w/each other.
6. Learn search trees, balanced search trees, and Hashing.
7. Learn basic graph algorithms and their applications
8. Learn basics of number theory and application to cryptography

Academic Integrity: Familiarize yourself with NJIT Honor Code:

<https://www.njit.edu/dos/academic-integrity> .

Any evidence of dishonesty will be dealt with seriously and reported to the Dean of Students.

Weekly Course Outline:

week	Topic	Reading
1	Introduction, Analysis Techniques Worst-Case and Average-Case Analysis Complexity Definitions: $O()$, $\Omega()$, $\Theta()$ Insertion Sort: Worst-case and average-case	Module 1: Analysis
2, 3	Proof by Induction, Recursive Algorithms, Divide-and-Conquer Mergesort Recurrence Equations & Master Theorem	Module 2 : Proofs Module 3 : Recursive Algorithms Module 4 : Recurrences
4, 5	Sorting Algorithms: Heapsort Quicksort: Worst-case and Average-case Integer Sorting: Bucket-Sort, Radix Sort Selection (K^{th} smallest element)	Module 5: Sorting and Selection
6, 7	Dictionary ADT (Search, Insert, Delete) 1 Binary Search Trees (BST); Average Analysis 2 Balanced Search Trees AVL Trees, Red-Black Trees 3 Hashing	Module 6: Search Trees & Hashing
8	Midterm Exam: Will cover modules 1-6	
9	Design Techniques: Divide-and-Conquer Long Integers Mult ; Strassen's Matrix Mult Greedy: TSP; Huffman Coding Dynamic Programming: Matrix-Chain	Module 7: Design Techniques
10,11	Graph Algorithms Definitions, Representations Traversals Single-Source-Shortest-Paths (Dijkstra) All-Pairs-Shortest-Paths (Floyd) MST Algorithms: Prim, Kruskal	Module 8: Graph Algorithms
	Advanced Graph Algorithms (Optional Study) Strongly Connected Components Biconnected Components Graph Matching	Module 9: Advanced Graph Alg (This module to be added in future.)
12,13	Number Theory and Cryptography	Module 10: Number Theory
14	Review	
	Final Exam (Will cover everything)	

Schedule of Assignments, Tentative Due Dates, and Exam Dates
(Additional assignments may be posted during the semester.)

Assignment	Study	Week Due	Due Date
Program 1: Insertion Sort	Module 1	Week 3	Fri Sep. 20
Homework 1: Analysis of Algorithms	Modules 1,2,3	Week 4	Fri Sep. 27
Homework 2: Recurrences; Heaps	Module 4	Week 5	Fri Oct. 4
Program 2: Sorting Algorithms	Module 5	Week 7	Fri Oct. 18
Midterm Exam (Class Time)		Week 8	Tues. Oct. 22
Program 3 (Radix LSD)	Module 5	Week 10	Fri Nov. 8
Homework 3 (Number Theory)	Module 10	Week 14	Fri Dec. 6
Final Exam			Date to be announced