CS 435 **Course Syllabus** **Spring 2023**	**Data Structures & Algorithms** **Sect 2: TR 1:00-2:20 KUPF 210** **Recitation F 1-2 CKB 217** **Sect 4: TR 4:00-5:20 GITC 1100** **Recitation F 4-5 GITC 1100**	**Prof. David Nassimi**

Prof.: David Nassimi Webex: njit.webex.com Email: nassimi@njit.edu Office: ITC 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:

In-Person (GITC 4308): T,R 2:30-3:45

Online Hour (WebEx): Wed 4:00-5:30
Meeting Number: 2623 516 4482
Link: https://njit.webex.com/njit/j.php?MTID=md0a1c4dc48175165ce501355a24b91e5

Webex class meeting (in case instruction is online)

To join the class meeting:

1. Log on WebEx using your UCID. (**This is a necessary first step.**)
2. Input meeting number or use the link.

(Warning: If you try to use the link directly, without first logging on webex, you end up waiting in the lobby.)

Section 2 Class (Starting 1 pm): Meeting Number: 2623 976 2731
Link: https://njit.webex.com/njit/j.php?MTID=m0ef9e8b9cd10f3099dfbb66be6c1f52e

Section 4 Class (Starting 4 pm): Meeting Number: 2624 953 2844
Link: https://njit.webex.com/njit/j.php?MTID=ma675a480e8f255339ca23d1ccebee0a0

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	30%
	Midterm Exam	35%
	Final	35%

Note: The average of your midterm and final exams must be at least 50 to pass the course.

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

Policies:

1. Assignments must be done by you individually (no team-work).
2. All assignments must be submitted electronically on Canvas by the due date.
3. Programming assignments must be in C, C++, JAVA, or Python.

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**: The TA needs to read this file to evaluate your program, and run the program to verify that it works. This is the file you used to run the program.
    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.

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 (Kth 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	
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	

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 2/3
Homework 1: Analysis of Algorithms	Modules 1,2,3	Week 4	Fri 2/10
Homework 2: Recurrences; Heaps	Module 4	Week 5	Fri 2/17
Program 2: Sorting Algorithms	Module 5	Week 7	Fri 3/3
Midterm Exam (Class Time)		Week 8	Thurs. March 9
Program 3 (Radix LSD)	Module 5	Week 10	Fri 3/31
Homework 3 (Number Theory)	Module 10	Week 14	Fri 4/28
Final Exam			Date to be announced