

CS-435-001 Syllabus

Catalog Description

Advanced topics in data structures and algorithms, involving sequences, sets, and graphs such as searching, sorting, order statistics, balanced search tree operations, hash tables, graph traversals, graph connectivity and path problems. Algebraic and numeric algorithms. Performance measures, analysis techniques, and complexity of such algorithms.

Student-friendly Description

Algorithms and data structures affect the lives of billions of people daily. Think about it this way: if you have ever used a cell phone then you have been a user of several algorithms that route calls, decode information, remove noise, sort data, or move graphics around fast. This course introduces undergraduate students to notions and techniques that are used as 'building blocks' in the design of algorithms.

We will study runtime analysis, correctness proofs and specific algorithms, using as motivation problems that appear in programming interviews. We will also discuss problems that seem to be impossible to be solved efficiently, posing one of the most challenging problems of our time.

Contact Information

Instructor: Ioannis Koutis

Email: ikoutis+cs435@njit.edu

Office: GITC 4314

Course Time and Location: Monday and Wednesday 11:30-12:50pm (KUPF203)

Office Hours: Monday 1:00-2:00, Wednesday: 3:00-4:00 and by appointment

Course Assistant: Qianwen Ye

Email: qy57@njit.edu

Office Hours: Thursday 4:00-5:00pm

Prerequisites and Textbook

Prerequisites: CS 241, CS 288

Textbook: T.C.Cormen, C.E.Leiserson, R.L.Rivest, and C. Stein. "Introduction to Algorithms", 3rd edition, MIT Press, ISBN-10 : 0262033844. We abbreviate it in class as *CLRS*.

Alternative Readings:

S. Skienna, "The algorithm design manual", Springer, ISBN-10: 1849967202S.

Dasgupta, C. Papadimitriou, U. Vazirani. "Algorithms", McGraw-Hill Education, ISBN-10: 0073523402.

Additional material such as lecture slides and notes will be posted on Moodle

Coursework and Evaluation

Homework Assignments [12%]. There will be 5 homework assignments. Only the best 4 assignment grades will be used in the weighted average, each contributing 3%. This means that you have the flexibility to skip one assignment. Try to make wise use of the option!

Mini-Project [18%]. There will be one nearly semester-long programming project. We will be slowly building the required components during the semester following certain milestones. The project will be announced after the first midterm. Evaluation will include a short presentation to the instructor.

Midterm Exam A [20%]. Open book and open notes. You can use any book and notes you bring into the exam. You are not allowed to use a computer or borrow notes from others.

Midterm Exam B [20%]. Open book and open notes. You can use any book and notes you bring into the exam. You are not allowed to use a computer or borrow notes from others.

Final Exam [30%]. Cumulative. The final exam will consist of two parts. Part 1 will be closed-books and notes, but questions will come directly from the homework assignments [10%]. Part 2 will be similar to midterms [20%].

Letter Grades. The conversion of raw grades will be based on grouping the raw grades into clusters using the k-means method, and then assigning a letter grade to each cluster. The letter grade assignment will be in accordance to the undergraduate grade legend (<https://www.njit.edu/registrar/policies/grading.php>). This implies that there is no limit on the percentage of students who can get any given letter grade (everyone can get an A).

Lateness Policy. Each student starts the semester with a **total budget** of 72 hours of combined delay. If the budget is exceeded, 2% will be subtracted from the delayed assignment for each hour of delay.

Tentative Course Schedule

Homework Dates:

Hw1: out on 09/10, due on 09/19

Hw2: out on 10/01, due on 10/10

Hw3: out on 10/15, due on 10/24

Hw4: out on 11/05, due on 11/14

Hw5: out on 11/19, due on 11/28

Miniproject: out on 10/01 due on 12/10

Exam Dates:

Midterm 1: 09/26

Midterm 2: 10/22

Final: (date and time set by registrar)

Any modifications will be done in consultation with the attending students and will be announced on Moodle.

Tentative Topics

1. The RAM model, Asymptotic Running Time Analysis, Recurrences, Brute Force Solutions, Proofs of Correctness.
2. Greedy Algorithms, Huffman Encoding, Binary Trees.
3. Problem Modeling and Representations. Graphs. Spanning Trees. Graph Traversals. Graph Decompositions. Heaps.

4. Statistics, Sorting, Selection. Divide-and-Conquer, Recursion. Randomness. Hashing.
 5. Dynamic Algorithms, Red-Black Trees, AVL Trees.
 6. Dynamic Programming, Shortest Path and Walk Counting Problems, Matrix Multiplication, Strassen's matrix multiplication.
 7. Problems with Graphs and Strings.
 8. NP-Hardness, Reductions, Coping with Hardness.
-

Course Policies

Email

Use of your NJIT email is strongly encouraged to avoid delivery problems.

Moodle Announcements

All course announcements will be posted on Moodle. So, check it regularly, or even better **subscribe** to the corresponding forums to receive e-mail notifications.

Moodle Discussion Forums

We will use Moodle to have informal and friendly conversations about topics related to the course, including assignments, problems, ideas, etc. You are encouraged to participate. Please be absolutely assured that any question or idea is welcome; even if you are afraid it may be silly, it can lead to interesting and beneficial discussions.

Mobile Devices

Turn-off and place away any cell phones, or other mobile devices, including laptops (unless the instructor explicitly permits you otherwise).

Assignment Delivery

Be careful to read and follow any formatting and delivery instructions that come with each assignment.

Grading

Written work will be graded for conciseness and correctness. Use formal arguments. Be brief and to the point and write clearly. Material covered in class and appearing in the relevant notes and chapters of the designated textbook can be used without proof.

Grade Corrections

Check the grades in course work and report errors promptly. Please try and resolve any issue within one week of the grade notification.

Absenteeism

If you miss a class, it's up to you to make up for lost time. Missing two exams leads to an automatic F in the course. If you miss one exam you **must contact** the Dean of Students (DOS) within 2 working days from the day the reason for the absence is lifted with all necessary documentation. If DOS approves, your missing exam grade will be set equal to the average of the non-missing exam grades.

Exam Conflicts

CS435 is a high-numbered required course. In case of multiple exams on a same day, this exam has priority even if it is the last exam of the day.

Incomplete

A grade of I (incomplete) is given in rare cases where work cannot be completed during the semester due to documented long-term illness or unexpected absence for other serious reasons. A student needs to be in good standing (i.e. passing the course before the absence) and receives a provisional I if there is no time to make up for the documented lost time; a letter (or email) with a timeline of what is needed to be done will be sent to the student. Note that for most cases an I

would be resolved within few days, not months and not the following semester! Not showing up in the final will probably get you an F rather than an I.

Collaboration and External Resources for Assignments

Some homework assignment problems will be challenging. You are advised to first try and solve all the problems **on your own**. For problems that persist you are welcome to talk to the course assistant or the instructor. You are also allowed to collaborate with your classmates and search for solutions online. But you should use such solutions only if you understand them completely (admitting that you don't understand something is way better than copying things you don't understand). Also make sure to give the appropriate credit and citation.

Collaboration in the mini-Project and Exams

Such collaboration is strictly prohibited. Any submitted code (even few lines) obtained through the Internet or otherwise, or is product of someone else's work or is common with another student submission, in the same or other section/course, risks severe punishment, as outlined by the University; all parties of such interaction receive automatically 0 and grade is lowered by one or two levels.

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*

Learning Outcomes

LO1. Be able to understand the input and output specifications of a problem, the relationship between input and output and identify, define, and describe the algorithmic requirements that formulate a solution for that problem. [ABET-1]

LO2. Be able to asymptotically compare functions using $o, O, \omega, \Omega, \Theta$, and be able to solve recurrences using the master, the iteration/recursion tree, and the substitution method.

LO3. Be able to understand fundamental algorithms and data structures for problems such as sorting, searching, selection, operations on numbers, polynomials and matrices, and graphs, and be able to trace their operations.

LO4. Be able to identify the performance characteristics of algorithms and data structures introduced in class, and be able to select among multiple solutions to meet desired needs. [ABET-1]

LO5. Be able to derive and describe the asymptotic performance of algorithms and data structures. [ABET1]

LO6. Be able to design and implement some of the data structures and algorithms introduced in class or augmentations of them following a given set of provided requirements and specifications. [ABET-2], [ABET-6]

LO7. Be able to communicate effectively and describe in writing algorithm operations and reason about their behavior and performance succinctly. [ABET-3]

LO8. Be able to compose more complex algorithms and data structures using as building blocks the fundamental algorithms, data structures, or algorithmic design techniques.