CS 430 – Introduction to Algorithms Syllabus Version 1.0

Gruia Calinescu Department of Computer Science Illinois Institute of Technology

Spring 2021

1 Course Summary

Welcome to CS 430 Introduction to Algorithms. In this course we study basic techniques for algorithm design. We also use basic analysis methodology of the complexity of algorithms, with worst case and average case bounds on time and space usage. For this, we use the "Big Oh", "Theta", and "Omega" notation.

We start with data structures, and their application for efficient algorithmic solutions. Then we concentrate on efficient sorting techniques, followed by general techniques such as Greedy, Divideand-Conquer, and Dynamic Programming. We continue with basic graph algorithms, followed by specific examples from string matching and computational geometry.

We will also discuss a practical case study, including identifying the situations when the theory of algorithms is necessary, and the search for an appropriate algorithm.

Finally, we look at the notion of NP-Completeness as a defining characteristic of the hardness of a problem.

2 Textbook

The textbook is *Introduction to Algorithms* (third edition) by Cormen, Leiserson, Rivest, and Stein, MIT Press, 2009. The first and second edition would be O.K. for learning, but not for assigned homework.

Our library has on-line access to the third edition of this book: login into my.iit.edu, click in the right-upper corner "nine-squares", select *Library e-resources*, then find *Books 24x7*, click on it and search for "Cormen".

3 Prerequisites

CS 330 and CS 331, or CS 401. To be precise, I assume familiarity with:

- 1. Elementary data structures: stacks, queues, arrays, linked lists and trees. Recursive algorithms.
- 2. Discrete structures: sets, trees, directed and undirected graphs.
- 3. From calculus: functions, polynomials, matrices, and logarithms.
- 4. "Big Oh", Ω and Θ notions regarding growth of functions.

4 Sections and Recitations

Lectures will have time for questions, and will be recorded. Lectures will contain quizzes, possibly to be solved in groups. Every student is invited to the interactive online lectures, scheduled almost each Monday and Wednesday, at 2PM.

Students in Sections V06 and V07 (the asynchronous sections) will have two extra homeworks to replace all the quizzes in their score.

Students in Section V04 must sign in for one of the recitation sessions. They are also expected to attend both the lectures and their recitation session, and take the quizzes. Missed quizzes for valid reasons (i.e. doctor's note) will be replaced by homework problems given to Sections V06 and V07 (the asynchronous sections).

Starting with February, the recitation of section CS430-S05 (28175) will be in-class only, no recording (the room assigned, WH113, is very large: 267 seats), The recitation of section CS430-S04 (52445) will be remote-only, with recording. The third section, CS 430-S06 (52446) is yet to be decided. If everyone agrees on either remote or in-class, we will keep it that way. If at least one student requests in-class and another requests on-line participation, we will keep S06 hybrid.

Recitations are meant to be interactive. During recitations, typically a TA will:

- answer questions regarding lectures, homeworks, project
- discuss solutions to homeworks, quizzes, and midterm (inviting students to present their solution as well). Also cover common errors with the purpose of avoiding repeating them
- possibly more examples than covered in the lecture
- possibly present a few algorithms without analysis (not midterm material)
- possibly take quizzes if there was not enough time during the lectures

5 Getting Help

All the office hours will be on Blackboard or Beacon. Instructor's office hours are Monday and Thursday 3:30-4:30 PM. For other appointments send email to calinescu@iit.edu. The TAs for this class are Shuichi Maruyama - who will guide you during the **recitations** Fridays -(smaruyam@hawk.iit.edu) with office hours Wednesday 3:30-4:30 PM, Si Wei (swei14@hawk.iit.edu) with office hours Tuesdays and Fridays 11-12, and Qitian Zeng (qzeng3@hawk.iit.edu) with office hours on Beacon only, Wednesdays 8-9 AM (9-10 AM after DST starts). and Saturday 8-9 PM (9-10 PM after DST starts), and TBD

Handouts (including this syllabus, homeworks, quizzes) will be available at http://www.cs.iit.edu/~cs430. Partial homework solutions will be posted on blackboard. Students are expected to read their email (the "official" email address we have on their file) every weekday. Announcements will also be posted on the web page.

6 Grading

The grading allocation is given below.

Homeworks	36%
Project	12%
Quizz Answers	12%
Midterm	15%
Final exam	25%

The midterm and the final exams are going to be individual and "real-time" on blackboard. Students will have access to the textbook but are not supposed to use any other help. The midterm will be on **March 8**, during class. The final exam will be held as scheduled by the Registrar's office, during the week **May 10-14**. Makeup exams will be only for emergency situations. The lowest quiz scores will be dropped (so don't ask for excuses, except for emergencies) so that exactly 12 quizzes count for the grade.

Six homeworks will be assigned, and are to be solved individually. Seek help from me if you are having any difficulties with the homework. Many homeworks and exam problems will ask for designing an algorithm to solve a typical problem. Unless specified otherwise in the description of the problem, the solution includes arguing correctness, and giving and justifying good upper bounds on the running time of the algorithm. Each of these two reasonings (for correctness and running time) will be worth circa 10% of the grade on that particular problem.

The penalty for late assignments is: 10% one lecture late and 20% one week late. No assignment will be accepted if more than one week late.

A composite score will be computed according to the scale above. The official grade will follow the following scale, which I might adjust by + or -2% to ensure consistency with the previos six semesters I offered the class (in total, there were circa 39 A's, 110 B's, 74 C's, 19 D's, and 9 E/F's). Be aware that graduate students cannot get a D and instead will get an E/F. There is no prescribed curve and you are not competing with each other.

Standard departmental policy regarding academic (dis)honesty applies. This includes https://web.iit.edu/student-affairs/handbook/fine-print/code-academic-honesty

In particular, homework solutions copied from the Internet are not allowed. If I have evidence that the work submitted is not your own, I will assign a score of zero on that particular assignment at the first occurence. This happened several times in the last five years. It may also be reported to academichonesty@iit.edu. The second occurence will be reported and an appropiate sanction will be applied after consultation with the office of the Associate Provost for Academic Affairs. Whenever in doubt, ask first if some action is allowed or not. Moreover, the students must submit the signed College of Science Academic Integrity Pledge before the first homework. The project will probably be a team effort, and guidelines regarding use of other resources will be included in the description of the project. For the programming part of the project, we require Java, C++, or Python.

7 Topics to be covered

- 1. Algorithm Analysis and Mathematical Background (Chapters 2, 3)
- 2. Heaps, Heapsort (Chapter 6)
- 3. Divide-and-Conquer method and Quicksort, Medians (Chapters 4, 7, 9.2)
- 4. Hash Tables (Chapter 11)
- 5. Binary Search and Red-Black Trees (Chapters 12, 13)
- 6. Data Structure for Disjoint Sets (Chapter 21)
- 7. The Greedy Method (Chapter 16)
- 8. Dynamic Programming (Chapter 15)
- 9. Elementary Graph Algorithms (Chapter 22)
- 10. Minimum Spanning Trees (Chapter 23)
- 11. Shortest Paths (Chapters 24 and 25)
- 12. Pattern Matching (Chapter 32, time permitting)
- 13. Convex Hulls (Chapter 33, time permitting)
- 14. NP-Completeness (Chapter 34)

8 Administrative Matters

Reasonable accommodations will be made for students with documented disabilities. In order to receive accommodations, students must obtain a letter of accommodation from the Center for Disability Resources and make an appointment to speak with me [the instructor] as soon as possible. The Center for Disability Resources (CDR) is located in Life Sciences Room 218, telephone 312-567-5744 or disabilities@iit.edu.

Illinois Techs Sexual Harassment and Discrimination Information:

Illinois Tech prohibits all sexual harassment, sexual misconduct, and gender discrimination by any member of our community. This includes harassment among students, staff, or faculty. Sexual harassment of a student by a faculty member or sexual harassment of an employee by a supervisor is particularly serious. Such conduct may easily create an intimidating, hostile, or offensive environment.

Illinois Tech encourages anyone experiencing sexual harassment or sexual misconduct to speak with the Office of Title IX Compliance for information on support options and the resolution process.

You can report sexual harassment electronically at iit.edu/incidentreport, which may be completed anonymously. You may additionally report by contacting the Title IX Coordinator, Virginia Foster at foster@iit.edu or the Deputy Title IX Coordinator at eespeland@iit.edu. For confidential support, you may reach Illinois Techs Confidential Advisor at (773) 907-1062. You can also contact a licensed practitioner in Illinois Techs Student Health and Wellness Center at verb+student.health@iit.edu+ or (312)567-7550

For a comprehensive list of resources regarding counseling services, medical assistance, legal assistance and visa and immigration services, you can visit the Office of Title IX Compliance website at https://www.iit.edu/title-ix/resources.

9 CS 430 Course Outcomes (as required for accreditation):

1. Use big O, omega, and theta notation to give asymptotic upper, lower, and tight bounds on time and space complexity of algorithms.

2. Determine the time complexity of simple algorithms, deduce the recurrence relations that describe the time complexity of recursively defined algorithms, and solve simple recurrence relations.

3. Design algorithms using the brute-force, greedy, dynamic programming, divide-and-conquer, branch and bound strategies.

4. Design algorithms using at least one other algorithmic strategy from the list of topics for this unit.

5. Use and implement the fundamental abstract data types – specifically including hash tables, binary search trees, and graphs – necessary to solve algorithmic problems efficiently.

6. Solve problems using techniques learned in the design of sequential search, binary search, O(N log N) sorting algorithms, and fundamental graph algorithms, including depth-first and breadth-first search, single-source and all-pairs shortest paths, and at least one minimum spanning tree algorithm.

7. Demonstrate the following abilities: to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in simple programming contexts.

8. Understand the notion of complexity, especially in the context of NP-Hard problems.

10 CS 430 Program Outcomes (as required for accreditation):

1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.

2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program?s discipline.

3. Communicate effectively in a variety of professional contexts.

7. Be prepared to enter a top-ranked graduate program in Computer Science