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, Divide-and-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.

All CS 430 students are expected to attend the recitation session (also known as lab) on Fridays 10:00-10:50.

2 Textbook


Our library has on-line access to the third edition of this book:

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 Getting Help

Office hours are Tuesday 2-3 PM and Thursday 3-4 PM, in room SB 228D (312-567-5273), or by appointment. I will also be available up to 15 minutes after the class, in the class. For an appointment send email to calinescu@iit.edu. You can also call me at 312-567-5273.

The TA for this class is Xiaolang Wang, who will guide you during the recitation hours Fridays, 10:00-10:50. Xiaolang (xwang122@hawk.iit.edu) has office hours Monday and Friday 1-2, in SB 004 (basement of Stuart Building), phone 312-567-5149.

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.

5 Grading

The grading allocation is given below.

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homeworks</td>
<td>30%</td>
</tr>
<tr>
<td>Project</td>
<td>15%</td>
</tr>
<tr>
<td>Class Participation or Quizz Answers</td>
<td>5%</td>
</tr>
<tr>
<td>Midterm</td>
<td>20%</td>
</tr>
<tr>
<td>Final exam</td>
<td>30%</td>
</tr>
</tbody>
</table>

The midterm and the final exams are closed books and closed notes (except for the notes supplied by the instructor), and will contain, among other problems, modified homework problems. The midterm will be on Oct. 18, in class. The final exam will be held as scheduled by the Registrar’s office, during the week Dec. 3-8. Makeup exams will be only for emergency situations.

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 problem 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 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.

Class participation is a subjective quantity, based mainly on how I feel you helped with comments and questions during the class. Just showing up gives you half of the points, but talking while I talk can decrease your score to zero.
During recitation, up to seven quizzes will be given. Limited collaboration and use of non-electronic materials will be allowed. The TA will then discuss solutions. Those whose solution is close to being correct, get one “Quiz Answer” point per quiz, up to a maximum of five such points.

The “Class Participation or Quizz Answers” quantity above is the maximum of the two scores. 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 previous five semesters I offered the class (in total, there were circa 36 A’s, 98 B’s 67 C’s 15 D’s and 8 E/F’s). There is no prescribed curve and you are not competing with each other.

<table>
<thead>
<tr>
<th>Score Range</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>88 - 100%</td>
<td>A</td>
</tr>
<tr>
<td>76 - 87 %</td>
<td>B</td>
</tr>
<tr>
<td>64 - 75%</td>
<td>C</td>
</tr>
<tr>
<td>52 - 63 %</td>
<td>D</td>
</tr>
<tr>
<td>0 - 51 %</td>
<td>E</td>
</tr>
</tbody>
</table>

Standard departmental policy regarding academic (dis)honesty applies. This includes [https://web.iit.edu/student-affairs/handbook/fine-print/code-academic-honesty](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 occurrence. This happened several times in the last five years. It may also be reported to academichonesty@iit.edu. The second occurrence will be reported and an appropriate 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 together with 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.

Those taking notes on laptops or other electronic devices (why else bring them in?): sit starting with the last row. Turn off all other electronic devices. 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.

6 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)

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

a. An ability to apply knowledge of computing and mathematics appropriate to the program’s student outcomes and to the discipline.
b. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
j. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
l. Be prepared to enter a top-ranked graduate program in Computer Science.