

A course on algorithms and data-structures. Methods for the analysis of algorithms are introduced, algorithms for sorting, searching, and selection, and data structures that support fast and efficient information retrieval are presented (hashing, heaps and priority queues with applications to data compression, binary search trees, red-black trees). Arithmetic algorithms. Greedy algorithms and dynamic programming-based techniques are introduced in the context of graph algorithms, arithmetic algorithms and text compression. Graph algorithms for traversals (depth-first, breadth-first), shortest-path problems, and spanning tree algorithms are also introduced. String matching algorithms. Introduction to NP-completeness.

1.1 Contact Information

INSTRUCTOR:	Alex Gerbessiotis	E-MAIL:	alg435@cs.njit.edu
OFFICE:	GITC 4213, 4th floor	TEL:	(973)-596-3244
OFFICE HOURS:	Mon 4:45-6:00pm and Thu 10:00-11:30am		
OFFICE HOURS:	By appointment Mon/Wed/Thu		
ASSISTANT:	TBA on course web-page		
CLASS HOURS:	Mon 2:30-3:55pm, Thu 4:00-5:25pm, Rm: FMH 404		
WEB PAGE:	http://www.cs.njit.edu/~alexg/courses/cs435/index.html		

Print Handout 1 from Web-page and compare the printout to this document! They must be identical.

1.2 Course Administration

Prerequisites CS 114, CS 241. (Math 226 has been phased out as a prerequisite and is marginally acceptable.)

Textbook T.C.Cormen, C.E.Leiserson, R.L.Rivest, and C. Stein. "Introduction to Algorithms", 2nd edition, McGraw-Hill, ISBN : 0-07-013151-1. *We abbreviate it in class as CLRS.* Note that a 3rd edition was published in Fall 2009; you might use it but the official textbook THIS SEMESTER is still the 2nd edition. All references in class/notes will be to the 2nd edition. Differences between the two are minor, and mainly on page numbers.

CourseWork: 4 exams (including the final); programming assignments.

Grades: 1000 points = PA(134) + Ex1(100) + Ex2(333) + Ex3(100) + Ex4(333).

PA1-3 3 programming assignments(PAs) will be handed out. Each one is worth 134 points. You get 134 points credit towards the final grade as long as you collect 134 or more points from any combination of the three PAs (you can do one, or two, or all, completely, or in part); 133 or fewer will gain you 0 points towards the final grade.

PA Dates Programs **MUST be received by email before midnight of the day** indicated in the calendar. Submit early, do not wait until the very end; you might get sick, you might get called for jury duty. The indicated date is the last day we must receive your submission, and we will acknowledge it promptly by email. Follow Handout 2 for compliant forms of an email submission. No late work is accepted since one programming assignment can still satisfy the programming assignment requirement. Plan ahead of time, and submit early.

Practice PS Approximately eight problem sets PS1-8 will be periodically posted along with their solutions. Exams 1 and 3 will draw from these problem sets for some or all of their problems.

Exams Dates in Course Calendar. Exam 1 is closed-everything. The other three exams are open-textbook only. You may bring a copy of the textbook but you are not allowed to borrow one during the exam. For the final, you may also bring in class a clean copy of Handout 5 on red-black trees in addition to the textbook. Exam1 is on **Thu Feb 18**, 45min, 100 points. Exam2 is on **Thu Mar 11**, 75min, 333 points. Exam3 is on **Thu Apr 15**, 45mins, 100 points. Exam4 is on **??? May ??**, 2hrs, 333 points (check with the Registrar).

Exam Conflicts This 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.**

2.1 Course Objectives and Outcomes

- Objective 1** Learn how to describe the asymptotic performance of algorithms and data structures.
- Objective 2** Learn how to derive and determine the asymptotic performance of algorithms and data structures.
- Objective 3** Learn how fundamental algorithms and data-structures operate, and understand their characteristics. Be able to choose among a variety of similar ones based on problem/program specification and requirements.
- Objective 4** Learn how to compose more complex algorithms using as building blocks the fundamental algorithms introduced in class.
- Objective 5** Learn how to compose more complex algorithms using the algorithmic design techniques introduced in class.
- Outcome 1** Be able to asymptotically compare functions using $o, O, \omega, \Omega, \Theta$.
- Outcome 2** Be able to solve recurrences using the master, the iteration/recursion tree, and the substitution method.
- Outcome 3** Become familiar with a variety of sorting algorithms and their performance characteristics (eg, running time, stability, space usage) and be able to choose the best one under a variety of requirements.
- Outcome 4** Be able to understand fundamental algorithms and data structures and be able to trace their operations for problems such as sorting, searching, selection, operations on numbers, polynomials and matrices, and graphs.
- Outcome 5** Be able to identify the performance characteristics of fundamental algorithms and data structures for problems such as sorting, searching, selection, operations on numbers, polynomials and matrices, and graphs.
- Outcome 6** Be able to understand fundamental algorithm design techniques and understand how to use them to solve a variety of algorithmic problems.
- Outcome 7** Be able to use the fundamental algorithms introduced in class to design algorithms for more complex problems and analyze their performance.
- Outcome 8** Be able to use the design techniques introduced in class to design algorithms for more complex problems and analyze their performance.

2.2 Topics to be covered

- T1 : AL1(1)/AL2(1)/AL3(1): Introduction, Algorithm Design Techniques (Incremental, Divide-and-Conquer)
- T2 : AL1(2)/AL2(1) : Sorting Algorithms (Insertion, Selection, BubbleSort, MergeSort) Asymptotic growth of functions
- T3 : AL1(3) : Recurrences
- T4 : AL3(3) : Brief Review on elementary data structures (Stacks,Queues,Trees,Lists)
- T5 : AL2(2),AL8(1) : HeapSort,PriorityQueues,Huffman Coding,and QuickSort(Worst-case and Average-case analysis)
- T6 : AL3(2),AL8(1) : Distribution-based sorting(Count/Radix/Bucket-Sort).Lower bounds on comparison-based sorting.
- T7 : AL3(3) : Selection in Linear Time.
- T8 : AL3(3) : Hashing, Balanced Binary Search Trees (Red-Black Trees).
- T9 : AL2(2),AL3(1) : Dynamic Programming and Chained Matrix Multiplication,Arithmetic problems
- T10: AL3(2),AL8(1) : Union Find Algorithms; Introduction to Graph Algorithms
- T11: AL2(1)AL3(2) : Depth First Search,Breadth First Search,Minimum Spanning Trees.
- T12: AL3(3) : Shortest path Algorithms (Dijkstra and Floyd-Warshall)
- T13: AL6(3) : NP-completeness.

Section 2.2 of the previous page contains a tentative list of topics that is intended to be covered in class. The code T_i refers to a topic. A topic may spread over one or more lectures. The code AL_i refers to the ACM Computing Curricula 2001 topic description code. In parentheses, we provide an approximate number of hours per topic. Hour coverage may change depending on circumstances (eg. class pace, weather). Minimum time requirements of the topics covered are. AL_1 (Basic algorithmic analysis):4, AL_2 (algorithmic strategies): 6, AL_3 (fundamental computing algorithms):12, AL_6 (the complexity classes P and NP) optional.

3.1 Tentative Course Calendar

Spring 2010					
Week**	Mon	Thu	PS with Solutions	PA	Comments
W1		1/21	PS1*	PA1-3out	
W2	1/25	1/28	PS2*		
W3	2/1	2/4	PS3*		
W5	2/8	2/11	PS4*		
W4	2/15	2/18+	Exam1-Thu	PA1in	An exam date is marked with +
W6	2/22	2/25	PS5*		
W7	3/1	3/4	PS6*		
W8	3/8	3/11+	Exam2-Thu	PA2in	
W-	3/15	3/18			Spring Recess; no classes
W9	3/22	3/25	PS7*		
W10	3/29	4/1			
W11	4/5	4/8	PS8*		
W12	4/12	4/15+	Exam3-Thu	PA3in	
W13	4/19	4/22			
W14	4/26	4/29			
W15	5/3				5/6 -5/12 is exam week
W16			Exam4		Check with Registrar for exam day

* Problem Sets (PS) with solutions are not for credit. ** In this calendar, a week starts on a Thursday

Any modifications or deviations from these dates, will be done in consultation with the attending students and will be posted on the course Web-page. It is imperative that students check the Course Web-page regularly and frequently.

3.2 Absenteeism

MISSING CLASS If you miss a class and there is no EXAM or PA due it's up to you to make up for lost time.

MISSING PA There are three scheduled programming assignments and only one is needed to collect the 134 points for the course. Plan ahead of time and submit early; do not wait until the last PA or the last moment to submit. No extensions are granted for any reason, medical or otherwise.

MISSING EXAM If you miss an exam and there is a valid documentation for your absence, such documentation must be presented within 3 working days from the day the reason for the absence is lifted. The maximum accommodation will be the number of missing days to the exam date. If you miss Exam1 or Exam3, then the Exam2 or Exam4 grade will be used respectively in such a case; no makeup will be provided for Exams 1 and 3.

Programs	Code must be ANSI compliant and neither hardware-specific nor OS-specific. Check Handout 2 for more information on the programming assignments. Programming problems are graded based on test instances decided by the grader on a test platform of the grader's choice (e.g. AFS). Do not expect partial credit if your code fails to run on all test instances unless you accompany your code submission with a detailed bug report.
Extensions	As stated earlier, no extension will be granted for the PAs for any reason. Therefore do no try to act opportunistically i.e skip PA1 and PA2 and aim at doing PA3 completely and correctly to collect 134 points. You might get sick for the period leading to PA3 or called for jury duty or something else might occur and we will grant you no extension for that, even if you have proper documentation.
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. Excess Programming points can be used to boost your course grade (we account them separately). DO NOT USE pencils to write down your solutions; if you decide to use a pencil do not complain about grading.
Grades	Check the marks in written work and report errors promptly. Resolve any issue related to Ex1-Ex3 and PA1-3 no later than the Reading Day. If you believe a grade you received for the solution of a problem is not representative of your effort, talk to the grader first and then to the instructor (if different). For programming assignments an email with your grade is sent back to you by replying to the email submission of the PA. The final grade is decided based on a 0 to 1000 point scale. A student who collects at least 500 points and completes the minimum programming requirements should expect a grade of C or better; 800 points or more are usually required for an A. The instructor reserves the right to push a student's grade up based on that student's significant programming effort.
Incomplete	A grade of I (incomplete) is given in rare cases where work can not be completed during the semester due to documented long-term illness or absence (e.g. unexpected national guard duty). A student needs to be in good standing (i.e. is passing the course otherwise before the absence and have satisfied programming requirements) and receives an I if there is no time to makeup for the documented lost time. Not showing up in the final will probably get you an F rather than an I.
Collaboration	Collaboration of any kind is NOT allowed in the in-class exams. Students who turn in code obtained through the Internet or otherwise, or is product of another person's/student's work, risk severe punishment, as outlined by the University. The work you submit must be the result of your own effort.
Mobile Devices	Mobile phones/devices and/or laptops/notebooks MUST BE SWITCHED OFF (NOT JUST SILENCED) before the class exams. Switch off noisy devices also before class.
Email/SPAM	Send email from an NJIT email address. NJIT spam filters or us will filter other email address origins. Do not send course email to the instructor's email address unless there is a good reason (e.g. you don't want the grader to read the email or it's urgent and you believe the instructor will respond faster). Include CS 435 in the subject line then. ■

The NJIT Honor Code will be upheld; any violations will be brought to the immediate attention of the Dean of Students. Read this handout carefully!