A course on algorithms and data-structures. 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. Greedy algorithms and dynamic programming-based techniques. String matching algorithms. Introduction to NP-completeness.

1.1 Contact Information

Instructor:	Alex Gerbessiotis	E-mail:	alexg+cs435@njit.edu
Office:	GITC 4213, 4th floor	Tel:	(973)-596-3244
Office Hours:	Wed 4:00-5:30pm and Thu 4:00-5:30pm	Else, by	appointment Mon/Thu
Assistant:	TBA on course web-page		
Class Hours:	Thu 6:00-9:05pm, CKB 220		

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 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 (ISBN-13: 978-0262033848). *We abbreviate it in class as CLRS.*

CourseWork: **3 exams (including the final)**; **Programming assignments** ; an unannounced quiz may be given (maybe after week 8) of 40 points; if i don't give you a quiz you get the points for free. If the Exam1 day is a snow day, then Exam1 will be held a week later; if that day is also a snow day it will be scheduled for April 14 as **Exam 2.5**, a 110pt or 150pt exam.

Grades: 1000 points = MP(160) + Ex1(110) + Ex2(345) + Ex3(345) + 40?

MP: A programming mini-project (MP) with 3 options each one worth 120 points. A student may submit one or two options but NOT more; both options must be submitted as one email. No more than 160 points of credit from the sum of the grades of the two options. If a student receives a total of 60 points or less, 0 will be recorded towards the final grade; this implies a threshold of 25% for clearing the mini-project hurdle. See Handout 2 for more details and how to submit it. All options are due BEFORE noon of **April 21, 2016**; emails received at noon time or later will have 40pts deducted per option, and so done at noon time the following day(s), if applicable (i.e. there are points to deduct).

PS: Approximately eight problem sets PS1-8 will be periodically posted along with their solutions. Exams may draw from these problem sets.

Exams: Dates in Course Calendar. Exam 1 (or 2.5) is closed-everything. The other exams (midterm and final) are open-textbook only. You may bring a hard-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 11**, 45min, 110 points. Exam2 is on **Thu Mar 3**, 2hrs, 345 points. Exam3 is on **Thu May 12**, 2hrs, 345 points.

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 structure operations.

Objective 2 Learn how to derive and determine the asymptotic performance of algorithms and data structure operations.

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 and data structures using as building blocks the fundamental algorithms and data structures 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 $o, O, \omega, \Omega, \Theta$, and be able to solve recurrences using the master, the iteration/recursion tree, and the substitution method.

Outcome 2 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 3 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 4 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; be able to select among multiple available solutions to meet desired needs.

Outcome 5 Be able to understand fundamental algorithm design techniques and understand how to use them to design, implement and evaluate a variety of algorithmic problems.

Outcome 6 Be able to use the fundamental algorithms introduced in class to design, implement and avaluate algorithms for more complex problems.

Outcome 7 Be able to use current algorithmic techniques and skills for computing practice.

2.2 Topics to be covered

```
T1 : AL1(2)/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(2),DS3(1)   : Recurrences
T4 : AL3(1),DS5(1)   : Brief Review on elementary data structures (Stacks,Queues,Trees,Lists)
T5 : AL2(2),AL7(1)   : HeapSort,PriorityQueues,Huffman Coding,and QuickSort(Worst-case and Average-case analysis)
T6 : AL3(2)          : Distribution-based sorting(Count/Radix/Bucket-Sort).Lower bounds on comparison-based sorting.
T7 : AL3(2)          : Selection. Selection in Linear Time.
T8 : AL3(4),AL7(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),AL7(1)   : Union Find Algorithms; Introduction to Graph Algorithms
T11: AL3(3)AL7(2)    : Depth First Search,Breadth First Search,Minimum Spanning Trees.
T12: AL3(3)          : Shortest path Algorithms (Dijkstra and Floyd-Warshall)
T13: AL4(2)          : 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 Ti refers to a topic. A topic may spread over one or more lectures. The code ALi refers to the ACM Computing Curricula 2013 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. AL1 (basic analysis):4, AL2 (algorithmic strategies): 6, AL3 (fundamental data structures and algorithms):12, AL4 (basic automata, computability and complexity) optional:2, AL7 (advanced data structures, algorithms and analysis): elective, DS3 (proof techniques):1, DS5(graphs and trees):1.

3.1 Tentative Course Calendar

Spring 2016				
Week**	Thu	PS with Solutions	MP	Comments
W1	1/21	PS1*	MP out	
W2	1/28	PS2*		
W3	2/4	PS3*		
W4	2/11	**Exam1**		
W5	2/18	PS4*		
W6	2/25	PS5*		
W7	3/3	**Exam2**		
W8	3/10			
W-	3/17		Spring Break	
W9	3/24	PS6*		
W10	3/31			
W11	4/7	PS7*		
W12	4/14	PS8*		
W13	4/21		MP in before noon	
W14	4/28			
W-	5/5			5/5 is a Reading Day
W15	5/12	**Exam3**		Final Examination

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

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

Makeup It's up to you to make up for lost time; no make up if we give the quiz, for any reason (including DOSS justified one). No MP extensions for any reason, medical or otherwise; you have 3 months to submit it. If you miss an exam and there is a valid documentation for your absence, such documentation must be presented to the Dean of Student Services (DOSS) 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 contact DOSS on the 3rd day and you are accommodated for just one, you will not be given a make-up. Let the instructor also know as soon as possible and simultaneously with DOSS.

Programs	Submitted code must conform to the requirements of Handout 2 and the MP. Code must be ANSI compliant and neither hardware-specific nor OS-specific or library-specific. Programming problems are graded based on test instances decided by the grader on a test platform of the grader's choice (e.g. afsconnect1.njit.edu). 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	None: you are given 3 months to submit the MP.
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. DO NOT USE pencils to write down your solutions; if you decide to use a pencil do not complain about grading AFTER AN/THE EXAM.
Grades	Check the marks in written work and report errors promptly. **Resolve any issue before the first Reading Day**, or for the case of Exam3 and MP, 7 calendar days from the day grades were posted/made available. 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 MP. **The final grade** is decided based on a 0 to 1000 point scale. Any student getting 25% (86 points) or less in the final exam will get an F, irrespective of other performance metrics. A student who otherwise collects at least 500 points and completes the minimum programming requirements should expect a grade of C or better; 850 points or more are usually required for an A but this threshold varies. 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 cannot 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. passing the course before the absence) and receives a provisional I if there is no time to makeup 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 withing 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	**Collaboration of any kind is NOT allowed in the in-class exams nor in the mini-project. Students who turn in code obtained through the Internet or otherwise, or is product of another person's/student's work (same or other section, same or other year, etc), risk severe punishment, as outlined by the University. The student will receive 0 in the MP and so will any collaborators (even unwitting ones). In addition to University penalties, the course grade will be reduced by one level (a C+ would become a C) for all parties involved. The work you submit must be the result of your own effort and you must safeguard it.**
Mobile Devices	Switch off (not just silence) mobile devices before class.
Email/SPAM	Always use NJIT email address; NJIT spam filters might be unpredictable. Include **cs435** in the subject line. Do not complain otherwise. ■.

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!