New Jersey Institute of Technology

A. V. Gerbessiotis | CIS 435

January 5, 2006 | Spring 2006

Course Syllabus: General Information | Handout 1

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). Greedy algorithms and dynamic programming-based techniques are introduced in the context of graph algorithms. Graph algorithms for traversals (depth-first, breadth-first), shortest-path problems, and spanning tree algorithms are also introduced. Introduction to NP-completeness.

Contact Information

Instructor:	Alex Gerbessiotis	E-mail:	alg435@cs.njit.edu
Office:	GITC 4213, 4th floor	Tel:	(973)-596-3244
Office Hours:	Tue 11:30-1pm, Mon and Tue 4:30-5:30pm		
Office Hours:	By appointment some other time on Mon,Tue,Wed		
Assistant:	TBA on course web-page		
Class Hours:	Tue 6-9pm, Room TBA		

Course Web Page: `http://www.cs.njit.edu/~alexg/courses/cis435/index.html`

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

Course Administration

Prerequisites	CIS 114, Math 226.
Textbook	T.C.Cormen, C.E.Leiserson, R.L.Rivest, and C. Stein. "Introduction to Algorithms", second edition, McGraw-Hill, ISBN : 0-07-013151-1. *We abbreviate in class this second edition as CLRS.*
CourseWork:	**4 exams (including the final). Programming assignments worth 240 points will be handed out with a minimum of 80 points collected**.
Grading scheme:	1000 points = Ex1 (100) + Ex2 (350) + Ex3 (200) + Ex4(350). If a student doesn't collect 80 programming points from the programming assignments, grade will be reduced by a half letter grade (i.e. an A becomes B+, but a C becomes a D, and a D becomes an F).
PA1-3	3 programming assignments will be handed out. Each one is worth 80 points. Students ARE REQUIRED to collect cumulatively at least 80 points from any combination of the assignments.
Practice PS	Seven problem sets PS1-7 will be periodically posted along with their solutions. Exams 1 and 3 will be based on these problem sets.
Exams	Dates in Course Calendar. Exam 1,3 are closed-everything. The other two exams are open-textbook only. For the final, you might bring in class a clean copy of Subject 11 on binary search trees in addition to the textbook. Exam1 is on **Feb 14**, 1hr, 100 points. Exam2 is on **Mar 7**, 2hrs, 350 points. Exam3 is on **Apr 11**, 1hr, 200 points. Exam4 is on **May 9**, 2hrs, 350 points.
Exam Conflicts	This is a high-numbered course. **In case of multiple exams on a same day, this exam has priority even if it is the last exam of the day**. If you can't make it on the final, drop the course; if you have made travel arrangements already, drop them.
Due Dates	Programs **MUST be received by email by midnight the day** they are due. No late work is accepted.

Tentative Course Calendar

Spring 2006				
Week	Thu	PS with Solutions	PA	Comments
W1	1/17	PS1*	PA1-3out	
W2	1/24	PS2*		
W3	1/31	PS3*		
W4	2/7	PS4*		
W5	2/14	**Exam1**	PA1in	
W6	2/21	PS5*		
W7	2/28			
W8	3/7	**Exam2**	PA2in	
W-	3/14	Break	Break	Break
W9	3/21			
W10	3/28	PS6*		
W11	4/4			
W12	4/11	**Exam3**	PA3in	
W13	4/18	PS7*		
W14	4/25			
W-	5/2			Friday Schedule!
W15	5/9	**Exam4**		

* Problem Sets (PS) with solutions are not for credit.

The following describes 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 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. AL1 (Basic algorithmic analysis):4, AL2 (algorithmic strategies): 6, AL3 (fundamental computing algorithms):12, AL6 (the complexity classes P and NP) optional.

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)       : Non comparison-based Sorting (Count-Sort, Radix-Sort, and Bucket-Sort).
                           Lower bounds on comparison-based sorting.
T7 : AL3(3)              : Selection. Selection in Linear Time.
T8 : Midterm.
T9 : AL3(3)              : Hashing, Balanced Binary Search Trees ( Red-Black Trees).
T10: AL2(2),AL3(1)       : Dynamic Programming and Chained Matrix Multiplication
T11: AL3(2),AL8(1)       : Union Find Algorithms; Introduction to Graph Algorithms
T12: AL2(1)AL3(2)        : Depth First Search,Breadth First Search,Minimum Spanning Trees.
T13: AL3(3)              : Shortest path Algorithms (Dijkstra and Floyd-Warshall)
T14: AL6(3)              : NP-completeness.
```

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.

Written Work DO NOT USE pencils to write down your solutions; if you decide to use a pencil do not complain about grading.

Programs Code must be ANSI compliant and compile on the test platform/compiler, otherwise the assigned grade will be 0. Check relevant handout for more information on the programming assignments.

Grading Written work will be graded for conciseness and correctness. Use formal arguments. Be brief and to the point. Label solutions with problem/subproblem number clearly. Programming problems will be graded based on test instances decided by the grader on a test platform (Windows PC or Unix machine) of his choice. Do not expect partial credit if your code fails to run on all test instances. Do not expect partial credit if your code does not compile. Excess Programming points can be used to boost your exam grade (we account them separately).

Extension policies **No extension** will be granted for the programming assignments for any reason as one needs to collect 80 of more than 200 total points.

Grade questions Check the marks in a written work and report errors promptly. **Make sure you report such problems to the grader or the instructor within two weeks from receipt** but **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).

Final Grade The final grade is decided based on the 0 to 1000 point performance with an adjustment made based on programming assignment performance. A student who collects at least 500 points should expect a passing grade (provided the minimum programming requirements are satisfied). The instructor reserves the right to push a student's grade up based on that student's quality of his/her programming effort.

Collaboration Students who turn in solutions (programming or otherwise) that are derived from solution outlines of past assignments/homeworks, were obtained through the Internet, or are a product of another student's work, risk severe punishment, as outlined by the University. The work you turn in MUST BE your own personal work, composed and written by you. If you talk a problem with a fellow student cite this clearly in your homework (name the fellow student before the solution of the problem in question). Your work will then be compared to the other student's work to verify that your solution was written by you and reflect your own personal effort. If you don't report it, it will be considered a violation of the course rules. You are not allowed to exchange code for the programming part of a homework. **Collaboration of any kind is NOT allowed in the in-class exams**. Open-textbook refers to the textbook by Cormen Leiserson Rivest and Stein. Students are not allowed to exchange textbooks, course-notes or anything else including erasers, pencils, calculators etc.

Mobile Devices Mobile phones/devices and/or laptops/notebooks MUST BE SWITCHED OFF before the class exams. Switch off noisy devices (eg mobile phones) before you enter the classroom for a lecture.

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 (eg. you don't want the grader to read the email). Include `CIS 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!