CSE 5311-502 (Advanced Algorithms) SYLLABUS

(Summer 1999: TTh 6:00-7:50, Preston Hall 102)

Instructor:	Bob Weems, Associate Professor (weems@cse.uta.edu)
Office:	341 NEB, 817/272-2337
Hours:	TTh 3:15-5:45
GTA:	TBA

Prerequisites: Algorithms & Data Structures (CSE 2320)
Theoretical Computer Science (CSE 3315)

Objectives: Deeper study of algorithms, data structures, and complexity classes.

Goals:
1. Exposure to more sophisticated analysis techniques, e.g. amortized complexity.
2. Exposure to specialized data structures and algorithms.
3. Exposure to models of algorithm design.

Textbook: Cormen, Leiserson, Rivest, ***Introduction to Algorithms***, MIT Press, 1990. (Henceforth known as CLR)

References:

S. Baase, ***Computer Algorithms, Introduction to Design and Analysis***, 2nd edition, Addison-Wesley, 1988.

R.L. Graham, D.E. Knuth, and O. Patashnik. ***Concrete Mathematics***, Addison-Wesley, 1989.

D. Gusfield. ***Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology***, Cambridge University Press, 1997.

D. Gusfield and R. Irving. ***The Stable Marriage Problem: Structure and Algorithms***, MIT Press, 1989.

E. Horowitz and S. Sahni. ***Fundamentals of Computer Algorithms***, Computer Science Press, 1978.

D.E. Knuth, ***The Art of Computer Programming***, Vols. 1 and 3, Addison-Wesley.

C.H. Papadimitriou. ***Computational Complexity***, Addison-Wesley, 1994.

R. Sedgewick, ***Algorithms***, 2nd ed., Addison-Wesley, 1988.

T. Standish, ***Data Structure Techniques***, Addison-Wesley, 1980.

C.J. Van Wyk, ***Data Structures and C Programs***, Addison-Wesley, 1988.

N. Wirth, ***Algorithms + Data Structures = Programs***, Prentice-Hall.

Exams: 2 exams, weighted equally. Each exam will be 50% closed notes, 50% open notes

Homework: Three assignments - NOT GRADED.

Project: A project should highlight a particular algorithm, but should also present the context (e.g. other algorithms). Implementation is not required, but could be useful in gaining insight. Some tired (or trivial) topics (e.g sequential sorting, convex hulls, FFT, traveling salesperson, LZ compression, elementary RSA cryptography, file structures from CSE 5330/5331) will NOT be acceptable. Likewise, be careful in pursuing topics outside of the ''bounds'' of the terms that appear in the index of CLR. (For example, genetic algorithms, neural computing, cache-coherence policies, network protocols, database join techniques, and computer chess-playing techniques are NOT appropriate.) Besides the write-up (nominally 15 pages), each student must give a 10-minute presentation (i.e. enough time for one example) on August 3 (6:00-10:00), August 4 (6:00-10:00), or August 5 (6:00-10:00).

Milestone	Due Date	Weight
Topic 1	July 6	10%
Topic 2	July 15	15%
Presentation	August 3-5	25%
Paper	August 5	50%

A topic submission will include 1) a list of resources, 2) a narrative explaining what you will do, and 3) copies of the primary resources. At least one of your primary resources must be from the last two years.

Recent books, conference proceedings (e.g. IEEE Foundations of Computer Science) or journals (*J.ACM*, *SIAM J. on Computing*, *Information Processing Letters*) or Springer-Verlag's extensive *Lecture Notes in Computer Science* will be useful resources in developing a topic. Time spent finding a good topic will reap benefits later . . .

I WILL NOT PROVIDE ASSISTANCE TO THOSE WHO HAVE NOT ALREADY TRIED ON THEIR OWN!
EXPECT TO SPEND A SATURDAY (IN THE LIBRARY) TO FIND A TOPIC.

Grade: Your grade will be based on the following weights:

Exams:	70% (Test 2: Tuesday, August 10, 6:00-7:50 pm)
Labs:	10% (Two labs, equal weight)
Project:	20%

Policies:

1. Attendance is not required, but is highly encouraged. Consult me in advance if you must miss class for a good reason.

2. You are expected to have at least skimmed the new material by the day we start that material in class. The material will be covered in the order given later.

3. A due date will be given for each homework. On the due date the solutions will be available in the copy center. The GTA will assist with homework questions.

4. CHEATING - YOU ARE EXPECTED TO KNOW UNIVERSITY POLICIES. All cases of plagiarism will be processed through University channels outside the CSE department. IN ALL CASES OF CHEATING I WILL RECOMMEND A FAILING COURSE GRADE!

5. Any request for special consideration must be appropriately documented **in advance**. (Special consideration does not include giving a higher grade than has been earned.)

6. Late lab/project documents are penalized 30% per day. After the due date neither I, nor the GTA will provide assistance.

7. Each student will have available *one* 48-hour no-penalty extension that may be applied to *one* of the lab assignments or *one* of the project documents. To use your extension you must send an email to the grader *before* the due time.

8. Phone calls. I will not answer my phone during office hours if someone is in my office. After the third ring the call is switched to the CSE office, so please leave a message with the secretary.
Electronic mail is preferable to phone calls.

Course Outline

Starred (*) topics are not in CLR

1. Mathematical Preliminaries
    - Recurrences - Master Method (4.3-4.4)
    - Probability (6.3-6.6)
    - Coupon collecting*
2. Selective Review and Extensions of Data Structures
    - AVL Trees*
    - Red-Black Trees - Review (14)
    - Augmenting Data Structures (15)
    - Amortized Analysis (18)
    - Self-Organizing Linear Search (Computing Surveys)*
    - Optimal Binary Search Trees (dynamic programming)*
    - Skip Lists*
    - Self-Adjusting Binary Search Trees (Splay trees/amortized analysis) (JACM)*
    - Priority Queues - Review (7)
    - Binomial Heaps (20)
    - Fibonacci Heaps (21)
    - Disjoint Sets (union-find trees) (22)
    - Hashing - Brent's Rehash*, Perfect Hashing*
3. Medians/Selection (10)
4. Graph Algorithms
    - Minimum Spanning Trees (24)
        - Brief review of Prim
        - Kruskal's Algorithm (application of union-find trees)
    - Max-Flow/Bipartite Matching (27.1-27.5)
        - Brief review of Ford-Fulkerson
        - Preflow-push methods
        - Vertex and edge connectivity
    - Depth-First Search
        - Biconnected Components
5. Stable Marriages* - Sedgewick handout, stable marriage lattice/rotations
    Introduction to stable roommates

TEST 1

6. String Matching
    - Knuth-Morris-Pratt Algorithm (34.4)
    - Rabin-Karp Algorithm (34.2)
    - Suffix Trees

7. Matrices
    Strassen's Matrix Multiplication (31.2)
    LUP decomposition (31.4)
8. Computational Geometry (35)
9. Intractability (36)
    Sample Intractable Problems
    Complexity Classes
    Reductions
    Polynomial-Time Approximation
10. Parallel Algorithms/P-Completeness*
TEST 2

Calendar/Topics

	June				July		
1	Syllabus/1.	3	1.			1	5.
8	2.	10	2.	6	6.	8	TEST 1
15	2./3.	17	4.	13	7.	15	8.
22	NO	24	CLASS	20	9.	22	
29	4.			27	10.	29	

	August		
3	PROJECT	5	PRESENTATIONS
10	TEST 2		

CSE 5311 Survey

(Please submit by the end of the second lecture)

Name:

Home Phone:

Work Phone (& Hours):

Other Courses this Semester:

Special Circumstances Affecting your Performance:

CS Related Experience/Undergraduate Degree:

(Optional) What do you hope to gain from this course?