CSE 2320-001: ALGORITHMS & DATA STRUCTURES

Fall 2007: TR 11:00-12:20, Nedderman 229

Instructor: Bob Weems, Associate Professor
Office: 344 Nedderman Hall (`weems@uta.edu`, `http://reptar.uta.edu`)
Hours: TR 1:00-2:30 pm

GTA: Senjuti Basu Roy
Office:
Email: `roy@cse.uta.edu`
Hours:

Prerequisites: C programming (CSE 1320)
Java programming (CSE 1325)
CSE 2315

Objectives: In future design situations, students will be capable of developing, applying, and evaluating algorithmic solutions.

Outcomes:
1. Understanding of classic approaches to algorithm design - decomposition, dynamic programming, and greedy methods.
2. Understanding of particular algorithms and data structures that have wide applicabilty.
3. Understanding of basic algorithm analysis concepts by applying math skills to worst-case and expected time using recurrences and asymptotic notation.
4. Improved programming skills - especially data structures, recursion, and graphs.

Textbook: R. Sedgewick, *Algorithms in Java*, *Parts 1-5, 3rd ed.*, Addison-Wesley, 2003.

References: S. Baase and A. Van Gelder, Computer Algorithms: *Introduction to Design and Analysis, 3rd ed.*, Addison-Wesley, 2000.

Cormen, Leiserson, Rivest, Stein, *Introduction to Algorithms, 2nd ed.,* MIT Press, 2001.

G.H. Gonnet and R. Baeza-Yates, *Handbook of Algorithms and Data Structures, 2nd ed.*, Addison-Wesley, 1991.

Readings: Indicated on calendar later in syllabus.

Homeworks: Six homeworks, with answers, are available on the course web page.

Grade: Based on the following weights:

Exams: 75% divided evenly among 3 exams.
Exam 3: December 11, 11:00-1:30

Programs: 15% divided evenly among three assignments.

ABET Outcome B (Experimentation) Assessment Project: 10%
(See Policy #12)

Policies:

1. Regular attendance is expected. You are expected to know lecture contents and announcements. I reserve the right to have surprise quizzes, each quiz being 2% of the semester grade taken from the 75% allocated to exams.

2. Lecture notes and sample code for various algorithms are on the course web page `http://reptar.uta.edu/NOTES2320/cse2320.html`.

3. You are expected to have read the assigned readings by the specified date. Lectures will review and augment the material, but will also consider exercises from the book.

4. CHEATING - YOU ARE EXPECTED TO KNOW UNIVERSITY POLICIES. If you are suspected of cheating, the matter must go through university channels outside of the CSE Department.

   a. **Academic Integrity Policy:** It is the policy of the University of Texas at Arlington to uphold and support standards of personal honesty and integrity for all students consistent with the goals of a community of scholars and students seeking knowledge and truth. Furthermore, it is the policy of the University to enforce these standards through fair and objective procedures governing instances of alleged dishonesty, cheating, and other academic/non-academic misconduct.

      You can assume responsibility in two ways. **First**, if you choose to take the risk associated with scholastic dishonesty and any other violation of the Code of Student Conduct and Discipline, you must assume responsibility for your behaviors and accept the consequences. In an academic community, the standards for integrity are high. **Second**, if you are aware of scholastic dishonesty and any other conduct violations on the part of others, you have the responsibility to report it to the professor or assistant dean of students/director of student judicial affairs. The decision to do so is another moral dilemna to be faced as you define who you are. Students who violate University rules on scholastic dishonesty are subject to disciplinary penalties, including the possibility of failure in the course and dismissal from the University. Since dishonesty harms the individual, all students, and the integrity of the University, policies on scholastic dishonesty will be strictly enforced.

   b. **Statement on Ethics, Professionalism, and Conduct of Engineering Students**: The statement is attached. Failure to sign the statement will result in 1) programming assignments to not be accepted, i.e. late penalty and 2) failure on exams.

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 programs are penalized according to the following schedule. LABS ARE DUE AT 10:45 AM ON THE DUE DATE, NOT MIDNIGHT. After the due time, assistance will not be provided.

Degree of lateness	Penalty
Up to 10:45 next day	10 pts
Up to 10:45 two days	30 pts
Up to 10:45 three days	60 pts

7. Each student will have available *one* 2-day, no-penalty extension that may be applied to *one* of the lab assignments. To use your extension you must send an email to a grader ***before*** the due time. An acknowledgement will be sent.

   RESUBMISSIONS BEFORE THE DUE TIME ARE PENALIZED 10 POINTS EACH. NO RESUBMISSIONS AFTER THE DUE TIME.

8. Each lab is graded as follows:

		Some Issues
a. Output/Code	60%	If you know that your program has problems, you should let the GTA know what parts are functional. Test cases that demonstrate the limited functionality are useful.
b. Internal Comments	6%	Identification at beginning of program, including the assignment, who you are, how to compile on OMEGA, and high-level description. Each function: identify each argument, describe processing, and each `return`. You may reference notes and text. Excess line-by-line comments are not needed, but the processing for each iteration of a (significant) loop should be explained.
c. Modularity	6%	Functions are used appropriately. `main()` is kept simple.
d. Structure	6%	Code is not unnecessarily complicated or long. It is often better to rewrite code rather than patching several times.
e. Names	6%	Should indicate the purpose of the function, variable/field, or type. Cute or misleading names will be penalized.
f. Spacing	6%	Indenting, blank lines, placement of {}. Be consistent.
g. Generality	10%	Program is not unnecessarily limited.

   All programs must be written in ***Java*** to compile and execute using a recent version of the JDK. Details for program submission will be included with each assignment.

   You are responsible for correctly sending each programming assignment to the GTA as an attachment. (cc: yourself)

   No points will be awarded for programs that do not compile. *Points for b-g will not be awarded to submissions that are not substantially complete and perform significant processing.*

9. GTA duties:

   a. Provide first-level of assistance for homeworks and labs.
   b. Grade programs and short-answer test problems.

10. Instructor duties:

    a. Lecture.
    b. Guidance
    c. Tests - preparation and grading long-answer test problems.
    d. Special consideration.
    e. Design homework and programming assignments.

11. Please email the following information to `roy@cse.uta.edu` by Thursday, September 6:

    a. Name.
    b. Additional email addresses.
    c. Special circumstances affecting your performance.
    d. (Optional) What do you hope to gain from this course?

12. Undergraduate students achieving a semester grade of C or better, but *failing* the experimentation assessment (below 60%) and *documenting* their circumstances will be assigned a semester grade of I (incomplete) and may re-attempt the assessment in the next semester. If the assessment is then passed, the semester grade will be changed from I to the achieved grade.

Course Content (in chronological order)

1. Algorithmic Concepts (1.1-1.3, 6.3-6.4, 5.2, 8.1-8.7, 2.6,12.4) - Disjoint Subsets, Selection Sort, Insertion Sort, Divide and Conquer, Mergesort (trivial recursion tree), Binary Search (with and without duplicates)
2. Growth of Functions (2.1-2.4, 2.6-2.7) - Asymptotic Notation (O, Ω, Θ), Upper Bounds, Lower Bounds
3. Summations - Geometric Series, Harmonic Series, Math Induction, Integrals
4. Recurrences (2.5) - Substitution Method, General Recursion Trees
5. Heapsort/Priority Queues (9.1-9.6, 11.4) - Properties, Building a Heap, Sorting, Integrating with Other Data Structures, Applications
6. Greedy Algorithms - Quality-of-Solution Issues, Unweighted Interval Scheduling, Knapsack, Huffman Codes
7. Dynamic Programming (5.3) - Weighted Interval Scheduling, Optimal Matrix Multiplication, Longest Common Subsequence, Longest Increasing Subsequence, Subset Sum, Knapsack

Exam 1: Items 1.-7.

8. Quicksort (7.1-7.8) - PARTITION (2 versions), Selection/Ranking
   Lower Bounds - Decision Tree Model, Stability (6.1)
   Counting (6.10) and Radix Sorts (10.1, 10.5)
9. Linked Lists (3.3, 2.6, 12.3, 3.5, 3.4) - Use in Dictionaries, Headers, Sentinels, Circular Lists, Double Linking
10. Stacks/Queues (4.2, 4.4, 18.1, 4.3, 4.5, 4.7) - Policies and Applications

11. Rooted Trees (5.4-5.7) - Structure, Traversals
    Binary Search Trees (12.6-12.9) - Properties, Operations
12. Balanced Binary Search Trees (13.3-13.4) - Structural Properties, Rotations, Insertions

Exam 2: Items 8.-12.

13. Hashing (14.1-14.4) - Concepts, Chaining, Open Addressing
14. Graph Representations (3.7, 17.3-17.4) - Adjacency Matrices, Adjacency Lists, Compressed Adjacency Lists
    Search - Breadth-First (5.8, 18.7), Depth-First (19.2, 5.8, 18.2-18.4)
    Search-Based Algorithms - Topological Sort (19.6), Strong Components (19.8)
15. Minimum Spanning Trees (20.1-20.3) - Three Versions of Prim's MST
16. Shortest Paths - Dijkstra's Algorithm (21.1-21.2), Warshall's Algorithm (19.3), Floyd-Warshall Algorithm (21.3)
17. Network Flows and Bipartite Matching (22.1, 22.2, 22.4) - Concepts, Augmenting Paths, Residual Network, Cuts, Max-flow Min-cut Theorem, Implementation, Performance Issues

Exam 3: Items 13.-17.

Calendar - with subject numbers from course content

August/September				October			
28	Syllabus	30	1.	2		4	8.
4		6	2.	9	Exam 1	11	9.
11	3.	13	4.	16	10.	18	11.
18		20	5.	23	???	25	12.
25	6.	27	7.	30	13.		

November				December			
		1	Exam 2				
6	14.	8		4		6	
13	15.	15	16.	11	Exam 3		
20		22	HOLIDAY				
27	17.	29					

November 2 is the last day to drop.