CSE 2320-001: ALGORITHMS & DATA STRUCTURES

Fall 2014: 13:00 - 13:50, ERB 130

Instructor: Yuanzhe Cai
Office: 502 ERB
Hours: MW 10:00 -11:30 a.m.

GTA: Contact information will be on the blackboard

Prerequisites: C programming (CSE 1320, including basic UNIX competence and acquaintance with data structures)
Discrete Structures (CSE 2315, including combinatorics, trees, and graphs)

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 C, 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, 3rd ed., MIT Press, 2009.

Readings: Indicated on calendar later in syllabus.

Grade: Based on the following weights:

Exams: Exam 1 (15%), Exam 2 (15%) and Exam 3 (30%)

4 Programs: 10% divided among four labs.

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 80% allocated to exams.

2. Lecture notes and sample code for various algorithms are on the blackboard.

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.
   http://www.uta.edu/conduct/

5. Any request for special consideration must be appropriately documented <u>in advance</u>. (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	5 pts
Up to 10:45 two days	15 pts
Up to 10:45 three days	30 pts
Up to 10:45 four days	60 pts

7. 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%	Beginning of file including `main()` should identify the assignment and who you are, along with giving a 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. Do not prompt for file names!

All programs must be written in ***C*** to compile and execute on `omega.uta.edu`. Details for program submission will be included with each assignment.

You are responsible for correctly submitting each programming assignment on Blackboard.

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.* Submissions not reflecting algorithmic problem-solving techniques will not receive credit.

8. GTA duties:

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

9. Instructor duties:

   a. Lecture.
   b. Guidance
   c. Tests – preparation tests.
   d. Special consideration.
   e. Design four labs.

10. If you require a reasonable accomodation for a disability, please contact me no later than the second week of this semester. Further details are available at http://www.uta.edu/disability

11. Occasional class-wide email messages (e.g. weather situations, clarifications) may be sent to the addresses recorded by MyMav. Messages will also be archived on the course web page.

Course Content (in chronological order)

1. Algorithmic Concepts (1.1-1.3) - Disjoint Subsets, Union Find algorithm
2. Growth of Functions (2.1-2.4, 2.6-2.7) - Asymptotic Notation (O, Ω, Θ), Upper Bounds, Lower Bounds
3. Linked Lists (3.3, 2.6, 12.3, 3.5, 3.4) - Use in Dictionaries, Headers, Sentinels, Circular Lists, Double Linking
4. Stacks/Queues (4.2, 4.4, 18.1, 4.3, 4.6) - Policies and Applications
5. Element Sort (6.1-6.3, 5.2, 8.1-8.7, 2.6) - Selection Sort, Insertion Sort, Divide and Conquer, Mergesort (trivial recursion tree)
6. 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)
7. Heapsort/Priority Queues (9.1-9.6) - Properties, Building a Heap, Sorting, Integrating with Other Data Structures

Exam 1: 1-7

8. Greedy Algorithms - Quality-of-Solution Issues, Unweighted Interval Scheduling, Knapsack, Huffman Codes
9. Dynamic Programming (5.3) - Weighted Interval Scheduling, Optimal Matrix Multiplication, Longest Common Subsequence, Longest Increasing Subsequence, Subset Sum, Knapsack/Memoization
10. Rooted Trees (5.4-5.7) - Structure, Traversals
11. Binary Search Tree (12.5-12.9) - Properties, Operations
12. Balanced Search Tree (13.3-13.4) - Structural Properties, Rotations, Insertions
13. Hashing (14.1-14.4) - Concepts, Chaining, Open Addressing

Exam 2: Items 8.-13.

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 (20.1-20.4) - Three Versions of Prim's MST, Kruskal'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 (Final Exam): Items 1.-17.

Calendar - with subject numbers from course content

August

		21	Syllabus
26	1.	28	

September

2	2.	4	3.
9	4.	11	
16	5.	18	6.
23	7.	25	
30	Exam 1		

October

		2	8.
7		9	9.
14	10.	16	
21	11.	23	12.
28	13.	30	

November

4		6	Exam 2
11	14.	13	
18	15.	20	16.
25	17.	27	Holiday

December

2	
9	Exam 3

October 29 is the last day to drop; submit requests to major advisor prior to 4:00 p.m.