CAS CS 330 - Spring 2013 - Introduction to Algorithms

Lecture meeting time: Tues/Thurs 11 - 12:30, CAS 222.

Lab meeting times: Mon 11-12 (GCB 201), Mon 12-1 (CAS 233) and Mon 3-4 PM (CAS 323B).

Class syllabus.

Instructor: Prof. John W. Byers
Email (preferred): byers @ cs . bu . edu

Phone: 617-353-8925
Office: MCS 270

Office Hours: Mon 9:30 - 11AM and Wed 3-4:30PM, held in MCS 270 (John's office).

Teaching Fellow: Sanaz Bahargam
Email: bahargam @ cs . bu . edu

Office Hours: Tues 4-5:30 and Wed 11-12:30, held in EMA 302 (next to ugrad lab).

Official Course Description: This course examines the basic principles of algorithm analysis; techniques of efficient programming; analysis of sorting and searching; graph algorithms; string-matching algorithms; matrix algorithms; integer and polynomial arithmetic; the fast Fourier transform; and NP-hard and NP-complete problems.

Textbook: Algorithm Design, by Kleinberg and Tardos. ISBN 0-321-29535-8.

Syllabus: The details of the class are available on the syllabus.

Homework Assignments and Other Handouts:

January 15: Assignment 1 (pdf), (latex source). Due at the beginning of class on January 31.

January 31: Assignment 2 (pdf), (latex source). Due at the beginning of class on February 14.

February 15: Assignment 3 (pdf), (latex source). Due at the beginning of class on February 28.

February 27: Sample midterm questions (pdf), (latex source).

March 5: Assignment 4 (pdf), (latex source). Due at the beginning of class on March 21.

March 21: Assignment 5 (pdf), (latex source). Due at the beginning of class on April 4.

April 5: Assignment 6 (pdf), (latex source). Due by 5PM on April 18.

April 18: Assignment 7 (pdf), (latex source). Due at the beginning of (the last day of) class, May 2.

May 1: Sample final questions (pdf), (latex source).

Lectures:

Lecture 1 (1/17): Review of syllabus. Stable matching problem. Gale-Shapley algorithm and implementation.
Kevin Wayne's lecture slides and demo, used with permission.

Lecture 2 (1/22): Five representative problems, followed by review material from Chapter 2 (slides). Begin Max sum subinterval problem solved three ways: O(n^3), O(n^2), O(n log n).

Lecture 3 (1/24): Finish max sum subinterval problem. Then start on intro to graphs, graph data structures, graph algorithms. Chapter 3 (slides).

Lecture 4 (1/29): Finish Chapter 3. BFS and connectivity. Testing for bipartiteness (algorithm developed in class by the students -- that was fun!). Directed graphs, DAGs, and topological sort.

Lecture 5 (1/31): Problem solving: Q7 in Chapter 3. Start in on Chapter 4: greedy algorithms (slides). Solving the interval scheduling and interval partitioning problems (demo).

Lecture 6 (2/5): Continuing greedy algorithms from Chapter 4: minimizing max lateness, Farthest-in-future algorithm for offline caching, Dijkstra's algorithm (+ demo, not covered in class).

Lecture 7 (2/7): Minimum spanning tree: properties and efficient algorithms (slides). Application to clustering.

Lecture 8 (2/12): Huffman codes (slides). Wrap-up greedy algorithms.

Lecture 9 (2/14): Start in on Chapter 5: divide-and-conquer. Review of basics, analyzing recurrences. Counting inversions (demo), finding the closest pair of points in 2-D (slides).

Lecture 10 (2/19): Problem solving: Q19 from Chapter 4. Fast integer multiplication. Start in on fast matrix multiplication. (slides).

Lecture 11 (2/21): Fast matrix multiplication, and start on fast polynomial multiplication via FFTs.

Lecture 12 (2/26): Finish up FFTs and begin Chapter 6: dynamic programming. (slides).

Lecture 13 (2/28): Problem solving: Q3 from Chapter 5. Then back to DP: weighted interval scheduling, start on segmented least squares problem.

Lecture 14 (3/5): Discussion of HW #3 problems. Problem solving: Q3 from Chapter 6. Finish segmented least squares problem and move on to Knapsack.

Lecture 15 (3/7): In-class midterm.

Week of 3/11: Spring Break -- no class.

Lecture 16 (3/19): Prof. Byers out of town for a PI meeting. Class cancelled.

Lecture 17 (3/21): Discussion of midterm answers. Problem solving: Q20 from Chapter 6. Segue back into Knapsack problem and solution in pseudo-polynomial time.

Lecture 18 (3/26): More advanced dynamic programs involving internal sequences: RNA Secondary Structure, and start of Sequence Alignment.

Lecture 19 (3/28): Sequence Alignment problem. Bellman-Ford shortest path algorithm, and its application to intradomain routing. (Slides).

Lecture 20 (4/2): Introduction to max-flow and min-cut problems, basic definitions, and Ford-Fulkerson algorithm. (Slides, (Demo).

Lecture 21 (4/4): Proof of the max-flow, min-cut theorem. Applications of max-flow: matchings in bipartite graphs, perfect matchings. (Slides).

Lecture 22 (4/9): More advanced applications of max-flow: edge-disjoint paths, project selection with prerequisites (Section 4.11). You are not responsible for material we did not cover in class.

Lecture 23 (4/11): Polynomial-time reductions, demonstrating hardness of problems. Problems and reductions: independent set, vertex cover, set cover, 3-SAT. (Slides).

Lecture 24 (4/16): P vs. NP. Definitions and elaboration. Notion of NP-Completeness. (Slides).

(4/18): No class, but Monday schedule, so regular lab.

Lecture 25 (4/23): Short digression on local search heuristics, since you'll need them to work on HW #7. We covered a handful of Chapter 12 slides, up through the Metropolis-Hastings method. Then, back to NP: Circuit-SAT as "the first" NP-Complete problem. Reduction from Circuit-SAT to 3-SAT. (Slides).

(4/25): Class cancelled.

Lecture 26 (4/30): Last group of NP-Completeness reductions: 3-SAT to Hamiltonian Cycle, TSP, and Subset Sum. Then, intro to approximation algorithms. (Slides).

Lecture 27 (5/2): Approximation algorithms: scheduling and center selection. Course wrap-up.

See the syllabus for a tentative gameplan of upcoming material.

Final Exam (Wednesday, 5/8, 12:30 - 2:30 PM), in the normal final exam slot for classes in our time block.
