5633 Analysis of Algorithms – Spring 06

2/1/06

Schedule

(subject to change)

Date	Material
Tu 1/17	**Analyzing algorithms (Ch. 2.2)** Best case and worst case runtimes; insertion sort, incremental algorithm
Th 1/19	**Asymptotic notation (Ch. 3, Ch. A)** O, Ω, Θ, o, limit-theorem; runtime for code-snippets *Homework 1 assigned*
Tu 1/24	**Heapsort (Ch. 6)** Abstract data types (ADT), priority queue, heap, heapsort, linear-time buildheap
Th 1/26	**Divide-and-conquer (Ch. 2.3) and recurrences (Ch. 4.1, 4.2)** Divide-and-conquer, merge sort, binary search; Runtime recurrences. Solving recurrences with recursion tree; solving the recurrence with the substitution method (induction) *Homework 1 due; homework 2 assigned*
Tu 1/31	**Master theorem (Ch. 4.3), more divide-and-conquer (Ch. 31.6 pages 879–880; Ch. 30 pages 822–824; 28.2)** Use of master theorem to solve recurrences. Repeated squaring for exponentiation, Fibonacci numbers, polynomial multiplication, Strassen's matrix multiplication.
Th 2/2	**Randomized algorithms (Ch. 5.1–5.3), random variables and expected values (Ch. C.3)** Hiring problem; Expected runtime analysis. Random variables, expected value. *Homework 2 due; homework 3 assigned*
Tu 2/7	**Quicksort (Ch. 7.1–7.4)** Quicksort, best-case and worst-case runtimes, randomized quicksort.
Th 2/9	**Sorting (Ch. 8.1, 8.2, 8.3)** Decision trees, lower $\Omega(n \log n)$ bound for comparison sorts, counting sort, radix sort *Homework 3 due; homework 4 assigned*
Tu 2/14	**Order statistics (Ch. 9)** Order statistics (find i-th smallest element); Randomized selection, deterministic selection in linear time
Tu 2/21	**Red-black trees (Ch. 13.1, 13.2, 13.3)** Red-black tree property, rotations, insertion; abstract data types, ADT dictionary *Homework 4 due; homework 5 assigned*
Th 2/23	**B-trees (Ch. 18.1, 18.2)** k-ary search trees, B-tree def., height, insertion
Tu 2/28	**Test 1** Material until 2/16 (inclusive)
Th 3/2	**Augmenting Data Structures (Ch. 14)** Augmenting red-black trees; Dynamic order statistics, interval trees
Tu 3/7	**Range Trees** Range trees, in 2 dimensions and in d dimensions; preprocessing time, query time. *Homework 5 due; homework 6 assigned*

Date	Material
Th 3/9	**Dynamic programming (Ch. 15.2, 15.3, 15.4)** Fibonacci, binomial coefficient, LCS: fill table, then construct solution from the table.
Tu 3/21	**Dynamic programming (Ch. 15.2, 15.3, 15.4)** Matrix chain multiplication; general outline of dynamic programming: Optimal substructure (recurrence), overlapping subproblems, fill table bottom-up or by memoization.
Th 3/23	**Greedy algorithms (Ch. 16.2 pages 380 middle – 384; problem 16-1 on page 402; Ch. 16.3)** Greedy algorithms (greedy-choice property, optimal substructure). Making change, fractional knapsack. Huffman codes *Homework 6 due; homework 7 assigned*
Tu 3/28	**Amortized analysis (Ch. 17.1, 17.2, 17.4)** Aggregate analysis (total runtime of n operations), accounting method (prepay for later operations); binary counter, dynamic tables
Th 3/30	**Union-Find (Ch. 21.1, 21.2, 21.3)** Operations, list implementation, tree implementation, union-by-weight / union-by rank, path compression. Ackermann function, and inverse Ackermann function α. *Homework 7 due; homework 8 assigned*
Tu 4/4	**Elementary Graph Algorithms (Ch. 22.1–22.4)** Representations of graphs, breadth-first search (BFS), depth-first search (DFS), topological sort
Th 4/6	**Minimum Spanning Trees (Ch. 23)** Prim (grows single tree), Kruskal (grows forest; uses union/find data structure) *Homework 8 due; homework 9 assigned*
Tu 4/11	**Test 2** Material from 2/21 until 3/30 (inclusive)
Th 4/13	**Single-source shortest paths (Ch. 24 without 24.4)** Optimal substructure, triangle inequality, relaxation step; Dijkstra (only for non-negative edge weights), predecessor tree (shortest path tree); Bellman-Ford, detection of negative-weight cycles; Shortest paths in a DAG
Tu 4/18	**All-Pairs Shortest Paths (Ch. 25.2)** Dynamic programming: Floyd-Warshall *Homework 9 due; homework 10 assigned*
Th 4/20	**P and NP (Ch. 34)** Decision problems, definition of classes P and NP, polynomial-time reductions
Tu 4/25	**P and NP (Ch. 34)** NP-hardness, NP-completeness; Show that problems are NP-complete by reducing from other problems; TSP, Clique, Independent Set, Vertex Cover, Hamilton Path, Hamilton Circuit *Homework 10 due; homework 11 assigned*
Th 4/27	**Maximum Flow (Ch. 26)** Flow networks; Max-flow min-cut, augmenting path, residual network
Tu 5/2	**Maximum Flow (Ch. 26)** Ford-Fulkerson, Edmonds-Karp *Homework 11 due*

The final exam will be on Saturday May 6th 8:00am – 10:45am.