5633 Analysis of Algorithms – Spring 10

2/16/10

# Schedule

(subject to change)

| Date | Material |
|---|---|
| Tu 1/12 | **Analyzing algorithms (Ch. 2.2)**<br>Best case and worst case runtimes; insertion sort, incremental algorithm |
| Th 1/14 | **Asymptotic notation (Ch. 3, Ch. A)**<br>$O$, $\Omega$, $\Theta$, $o$, limit-theorem; runtime for code-snippets<br>*Homework 1 assigned* |
| Tu 1/19 | **Divide-and-conquer (Ch. 2.3) and recurrences (Ch. 4.1, 4.2)**<br>Divide-and-conquer, merge sort, binary search; Runtime recurrences. Solving recurrences with recursion tree; solving the recurrence with the substitution method (induction) |
| Th 1/21 | **Master theorem (Ch. 4.3), more divide-and-conquer (Ch. 31.6 pages 879–880; Ch. 30 pages 822–824; 28.2)**<br>Use of master theorem to solve recurrences. Repeated squaring for exponentiation, Fibonacci numbers, polynomial multiplication, Strassen's matrix multiplication.<br>*Homework 1 due; homework 2 assigned* |
| Tu 1/26 | **Randomized algorithms (Ch. 5.1–5.3), random variables and expected values (Ch. C.3)**<br>Hiring problem; Expected runtime analysis. Random variables, expected value. |
| Th 1/28 | **Quicksort (Ch. 7.1–7.4)**<br>Quicksort, best-case and worst-case runtimes, randomized quicksort.<br>*Homework 2 due; homework 3 assigned* |
| Tu 2/2 | **Sorting (Ch. 8.1, 8.2, 8.3)**<br>Decision trees, lower $\Omega(n \log n)$ bound for comparison sorts, counting sort, radix sort |
| Th 2/4 | **Order statistics (Ch. 9)**<br>Order statistics (find $i$-th smallest element); Randomized selection, deterministic selection in linear time<br>*Homework 3 due; homework 4 assigned* |
| Tu 2/9 | **Red-black trees (Ch. 13.1, 13.2, 13.3)**<br>Red-black tree property, rotations, insertion; abstract data types, ADT dictionary |
| Th 2/11 | **B-trees (Ch. 18.1, 18.2)**<br>k-ary search trees, B-tree def., height, insertion<br>*Homework 4 due; homework 5 assigned* |
| Tu 2/16 | **Augmenting Data Structures (Ch. 14, +)**<br>Augmenting red-black trees; Dynamic order statistics, interval trees; range trees, in 2 dimensions and in $d$ dimensions; preprocessing time, query time. |
| Th 2/18 | **No class** |
| Tu 2/23 | **Dynamic programming (Ch. 15.2, 15.3, 15.4)**<br>Fibonacci, binomial coefficient, LCS: fill table, then construct solution from the table.<br>*Homework 5 due* |
| Th 2/25 | **Dynamic programming (Ch. 15.2, 15.3, 15.4)**<br>Matrix chain multiplication; general outline of dynamic programming: Optimal substructure (recurrence), overlapping subproblems, fill table bottom-up or by memoization.<br>*Homework 6 assigned* |

| Date | Material |
|---|---|
| Tu 3/2 | **Greedy algorithms (Ch. 16.2, problem 16-1 on page 402; Ch. 16.3)**<br>Greedy algorithms (greedy-choice property, optimal substructure). Making change, fractional knapsack. Huffman codes |
| Th 3/4 | **Midterm Review**<br>Review for the midterm exam<br>*Homework 6 due* |
| Tu 3/9 | **Midterm Exam**<br>Material from 1/12 until 2/25 (inclusive) |
| Th 3/11 | **Amortized analysis (Ch. 17.1, 17.2, 17.4)**<br>Aggregate analysis (total runtime of $n$ operations), accounting method (prepay for later operations); binary counter, dynamic tables |
| Tu 3/16 | **SPRING BREAK** |
| Th 3/18 | **SPRING BREAK** |
| Mo 3/22 | **Drop deadline to drop with a 'W'** |
| Tu 3/23 | **Union-Find (Ch. 21.1, 21.2, 21.3)**<br>Operations, list implementation, tree implementation, union-by-weight / union-by rank, path compression. Ackermann function, and inverse Ackermann function $\alpha$. |
| Th 3/25 | **Elementary Graph Algorithms (Ch. 22.1–22.4)**<br>Representations of graphs, breadth-first search (BFS), depth-first search (DFS), topological sort<br>*Homework 7 assigned* |
| Tu 3/30 | **Minimum Spanning Trees (Ch. 23)**<br>Prim (grows single tree), Kruskal (grows forest; uses union/find data structure) |
| Th 4/1 | **Single-source shortest paths (Ch. 24 without 24.4)**<br>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<br>*Homework 7 due; homework 8 assigned* |
| Tu 4/6 | **All-Pairs Shortest Paths (Ch. 25.2)**<br>Dynamic programming: Floyd-Warshall |
| Th 4/8 | **P and NP (Ch. 34)**<br>Decision problems, definition of classes P and NP, polynomial-time reductions<br>*Homework 8 due; homework 9 assigned* |
| Tu 4/13 | **P and NP (Ch. 34)**<br>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 |
| Th 4/15 | **Approximation Algorithms (Ch. 35.1 and 35.2)**<br>Constant factor approximation; Vertex cover; MST for Euclidean TSP<br>*Homework 9 due; homework 10 assigned* |
| Tu 4/20 | **Maximum Flow (Ch. 26)**<br>Flow networks; Max-flow min-cut, augmenting path, residual network |
| Th 4/22 | **Maximum Flow (Ch. 26)**<br>Ford-Fulkerson, Edmonds-Karp<br>*Homework 10 due* |
| Tu 4/27 | **Final Review**<br>Review for the final exam |

Chapter numbers refer to the CLRS book. "+" indicates additional material.

**The comprehensive final exam will be on Tuesday May 4th, 8pm – 10:30pm in the classroom.**