3343 Analysis of Algorithms – Fall 11

# Schedule

11/3/11

(subject to change)

| Date | Material |
|---|---|
| Th 8/25 | **Analyzing algorithms (Ch. 2.2)**<br>Best case and worst case runtimes; insertion sort, incremental algorithm<br>**No recitation** |
| Tu 8/30 | **Asymptotic notation (Ch. 3, Ch. A)**<br>$O$, $\Omega$, $\Theta$, $o$, limit-theorem; runtime for code-snippets<br>*Homework 1 assigned* **Recitation: Homework 1** |
| Th 9/1 | **Asymptotic notation (Ch. 3, Ch. A)**<br>$O$, $\Omega$, $\Theta$, $o$, limit-theorem; runtime for code-snippets **Recitation: Homework 1** |
| Tu 9/6 | **Heapsort (Ch. 6)**<br>Abstract data types (ADT), priority queue, heap, heapsort, linear-time buildheap<br>*Homework 1 due; homework 2 assigned* **Recitation: Homework 2** |
| Th 9/8 | **Recursion trees and induction (+)**<br>Recursive algorithms. Guess solution of recurrence using recursion trees and prove the correctness of the solution using induction. **Recitation: Homework 2** |
| Tu 9/13 | **Divide-and-conquer (Ch. 2.3) and recurrences (Ch. 4.3, 4.4)**<br>Divide-and-conquer, merge sort, binary search; Runtime recurrences. Big-Oh induction (substitution method)<br>*Homework 2 due; homework 3 assigned* **Recitation: Homework 3** |
| Th 9/15 | **Master theorem (Ch. 4.5)**<br>Use of master theorem to solve recurrences. **Recitation: Homework 3** |
| Tu 9/20 | **More divide-and-conquer (Ch. 31.6 pages 956–957; 4.2)**<br>Repeated squaring for exponentiation, Strassen's matrix multiplication.<br>*Homework 3 due; homework 4 assigned; project 1 assigned* **Recitation: Homework 4** |
| Th 9/22 | **Probability, random variables and expected values (Ch. C.2, C.3)**<br>Probability, random variables, expected values. **Recitation: Homework 4** |
| Tu 9/27 | **Randomized algorithms (Ch. 5.1–5.3)**<br>Hiring problem; Expected runtime analysis.<br>*Homework 4 due; homework 5 assigned* **Recitation: Homework 5** |
| Th 9/29 | **Quicksort (Ch. 7.1–7.4)**<br>Quicksort, best-case and worst-case runtimes, randomized quicksort.<br>**Recitation: Homework 5** |
| Tu 10/4 | **Sorting (Ch. 8.1, 8.2, 8.3)**<br>Decision trees, lower $\Omega(n \log n)$ bound for comparison sorts, counting sort, radix sort<br>*Homework 5 due; homework 6 assigned; project 1 due* **Recitation: Review Test 1** |
| Th 10/6 | **Order statistics (Ch. 9)**<br>Order statistics (find $i$-th smallest element); Randomized selection, deterministic selection in linear time. **Recitation: Review Test 1** |
| Tu 10/11 | **Test 1**<br>Material until 9/27 (inclusive) **Recitation: Homework 6** |
| Th 10/13 | **Red-black trees (Ch. 13.1, 13.2, 13.3)**<br>Red-black tree property, rotations, insertion; abstract data types, ADT dictionary<br>**Recitation: Homework 6** |
| Tu 10/18 | **B-trees (Ch. 18.1, 18.2)**<br>$k$-ary search trees, B-tree def., height, insertion<br>*Homework 6 due; homework 7 assigned; project 2 assigned* **Recitation: Homework 7** |

| Date | Material |
|---|---|
| Th 10/20 | **Dynamic programming (Ch. 15.4, +)**<br>Fibonacci, binomial coefficient, LCS: fill table, then construct solution from the table. **Recitation: Homework 7** |
| Tu 10/25 | **Dynamic programming (Ch. 15.3, 15.4., 16.2, +)**<br>0-1 Knapsack; general outline of dynamic programming: Optimal substructure (recurrence), overlapping subproblems, fill table bottom-up or by memoization.<br>*Homework 7 due; homework 8 assigned* **Recitation: Homework 8** |
| Th 10/27 | **Greedy algorithms (Ch. 16.2, problem 16-1 on page 402)**<br>Greedy algorithms (greedy-choice property, optimal substructure). Making change, fractional knapsack. **Recitation: Homework 8** |
| Tu 11/1 | **Elementary Graph Algorithms (Ch. 22.1–22.2)**<br>Representations of graphs, breadth-first search (BFS)<br>*Homework 8 due; homework 9 assigned* **Recitation: Homework 9** |
| Th 11/3 | **Elementary Graph Algorithms (Ch. 22.3–22.4)**<br>Depth-first search (DFS), topological sort; *Project 2 due* **Recitation: Homework 9** |
| Tu 11/8 | **Review for Test 2**<br>Review for test 2<br>*Homework 9 due* **No recitation** |
| Th 11/10 | **Minimum Spanning Trees (Ch. 23)**<br>Prim (grows single tree), Kruskal (grows forest; uses union/find data structure)<br>*Homework 10 assigned; project 3 assigned* **No recitation** |
| Tu 11/15 | **Test 2**<br>Material from 9/29 until 10/27 (inclusive)<br>**Recitation: Homework 10** |
| Th 11/17 | **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. **Recitation: Homework 10** |
| Tu 11/22 | **All-Pairs Shortest Paths (Ch. 25.2)**<br>Dynamic programming: Floyd-Warshall<br>*Homework 10 due; homework 11 assigned* **No recitation** |
| Th 11/24 | **Thanksgiving**<br>No class |
| Tu 11/29 | **P and NP (Ch. 34)**<br>Decision problems, definition of classes P and NP, polynomial-time reductions, NP-hardness, NP-completeness; Show that problems are NP-complete by reducing from other problems<br>**Recitation: Homework 11** |
| Th 12/1 | **P and NP (Ch. 34)**<br>TSP, Clique, Independent Set, Vertex Cover, Hamilton Path, Hamilton Circuit<br>**Recitation: Homework 11** |
| Tu 12/6 | **Review for Final Exam**<br>Review for final exam<br>*Homework 11 due; project 3 due* **No recitation** |

Chapter numbers refer to the CLRS book, 3rd edition.
"+" indicates additional material.

**The comprehensive final exam will be on Thursday 12/15, 10:30am – 1pm.**