

Schedule

(subject to change)

| Date | Material |
|---------|--|
| Tu 1/12 | Analyzing algorithms (Ch. 2.2) Best case and worst case runtimes; insertion sort, incremental algorithm |
| Th 1/14 | Asymptotic notation (Ch. 3, Ch. A) O , Ω , Θ , o , limit-theorem; runtime for code-snippets <i>Homework 1 assigned</i> |
| Tu 1/19 | 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) |
| 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) Use of master theorem to solve recurrences. Repeated squaring for exponentiation, Fibonacci numbers, polynomial multiplication, Strassen’s matrix multiplication. <i>Homework 1 due; homework 2 assigned</i> |
| Tu 1/26 | Randomized algorithms (Ch. 5.1–5.3), random variables and expected values (Ch. C.3) Hiring problem; Expected runtime analysis. Random variables, expected value. |
| Th 1/28 | Quicksort (Ch. 7.1–7.4) Quicksort, best-case and worst-case runtimes, randomized quicksort. <i>Homework 2 due; homework 3 assigned</i> |
| Tu 2/2 | Sorting (Ch. 8.1, 8.2, 8.3) Decision trees, lower $\Omega(n \log n)$ bound for comparison sorts, counting sort, radix sort |
| Th 2/4 | Order statistics (Ch. 9) Order statistics (find i -th smallest element); Randomized selection, deterministic selection in linear time <i>Homework 3 due; homework 4 assigned</i> |
| Tu 2/9 | Red-black trees (Ch. 13.1, 13.2, 13.3) Red-black tree property, rotations, insertion; abstract data types, ADT dictionary |
| Th 2/11 | B-trees (Ch. 18.1, 18.2) k-ary search trees, B-tree def., height, insertion <i>Homework 4 due; homework 5 assigned</i> |
| Tu 2/16 | Augmenting Data Structures (Ch. 14) Augmenting red-black trees; Dynamic order statistics, interval trees |
| Th 2/18 | Range Trees (+) Range trees, in 2 dimensions and in d dimensions; preprocessing time, query time. <i>Homework 5 due; homework 6 assigned</i> |
| Tu 2/23 | Dynamic programming (Ch. 15.2, 15.3, 15.4) Fibonacci, binomial coefficient, LCS: fill table, then construct solution from the table. |
| Th 2/25 | 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. <i>Homework 6 due; homework 7 assigned</i> |

| Date | Material |
|---------|---|
| Tu 3/2 | Greedy algorithms (Ch. 16.2, problem 16-1 on page 402; Ch. 16.3) Greedy algorithms (greedy-choice property, optimal substructure). Making change, fractional knapsack. Huffman codes |
| Th 3/4 | Midterm Review Review for the midterm exam <i>Homework 7 due</i> |
| Tu 3/9 | Midterm Exam Material from 1/12 until 2/25 (inclusive) |
| Th 3/11 | 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 |
| 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) Operations, list implementation, tree implementation, union-by-weight / union-by-rank, path compression. Ackermann function, and inverse Ackermann function α . |
| Th 3/25 | Elementary Graph Algorithms (Ch. 22.1–22.4) Representations of graphs, breadth-first search (BFS), depth-first search (DFS), topological sort <i>Homework 8 assigned</i> |
| Tu 3/30 | Minimum Spanning Trees (Ch. 23) Prim (grows single tree), Kruskal (grows forest; uses union/find data structure) |
| Th 4/1 | 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 <i>Homework 8 due; homework 9 assigned</i> |
| Tu 4/6 | All-Pairs Shortest Paths (Ch. 25.2) Dynamic programming: Floyd-Warshall |
| Th 4/8 | P and NP (Ch. 34) Decision problems, definition of classes P and NP, polynomial-time reductions <i>Homework 9 due; homework 10 assigned</i> |
| Tu 4/13 | 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 |
| Th 4/15 | Approximation Algorithms (Ch. 35.1 and 35.2) Constant factor approximation; Vertex cover; MST for Euclidean TSP <i>Homework 10 due; homework 11 assigned</i> |
| Tu 4/20 | Maximum Flow (Ch. 26) Flow networks; Max-flow min-cut, augmenting path, residual network |
| Th 4/22 | Maximum Flow (Ch. 26) Ford-Fulkerson, Edmonds-Karp <i>Homework 11 due</i> |
| Tu 4/27 | Final Review 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.