

# Schedule

(subject to change)

Date	Material
Tu 1/13	<b>Analyzing algorithms (Ch. 2.2)</b> Best case and worst case runtimes; insertion sort, incremental algorithm
Th 1/15	<b>Asymptotic notation (Ch. 3, Ch. A)</b> $O$ , $\Omega$ , $\Theta$ , $o$ , limit-theorem; runtime for code-snippets
Tu 1/20	<b>Asymptotic notation (Ch. 3, Ch. A)</b> $O$ , $\Omega$ , $\Theta$ , $o$ , limit-theorem; runtime for code-snippets <i>Homework 1 assigned</i>
Th 1/22	<b>Heapsort (Ch. 6)</b> Abstract data types (ADT), priority queue, heap, heapsort, linear-time buildheap
Tu 1/27	<b>Recursion trees and induction (+)</b> Recursive algorithms. Guess solution of recurrence using recursion trees and prove the correctness of the solution using induction. <i>Homework 1 due; homework 2 assigned</i>
Th 1/29	<b>Divide-and-conquer (Ch. 2.3) and recurrences (Ch. 4.1, 4.2)</b> Divide-and-conquer, merge sort, binary search; Runtime recurrences. Big-Oh induction (substitution method)
Tu 2/3	<b>Master theorem (Ch. 4.3)</b> Use of master theorem to solve recurrences. <i>Homework 2 due; homework 3 assigned</i>
Th 2/5	<b>More divide-and-conquer (Ch. 31.6 pages 879–880; 28.2)</b> Repeated squaring for exponentiation, Strassen's matrix multiplication. <i>Programming project 1 assigned</i>
Tu 2/10	<b>Probability, random variables and expected values (Ch. C.2, C.3)</b> Probability, random variables, expected values. <i>Homework 3 due; homework 4 assigned</i>
Th 2/12	<b>Randomized algorithms (Ch. 5.1–5.3)</b> Hiring problem; Expected runtime analysis.
Tu 2/17	<b>Quicksort (Ch. 7.1–7.4)</b> Quicksort, best-case and worst-case runtimes, randomized quicksort. <i>Homework 4 due</i>
Th 2/19	<b>Test 1</b> Material until 2/10 (inclusive)
Tu 2/24	<b>Sorting (Ch. 8.1, 8.2, 8.3)</b> Decision trees, lower $\Omega(n \log n)$ bound for comparison sorts, counting sort, radix sort <i>Homework 5 assigned</i>
Th 2/26	<b>Sorting (Ch. 8.1, 8.2, 8.3)</b> Decision trees, lower $\Omega(n \log n)$ bound for comparison sorts, counting sort, radix sort
Tu 3/3	<b>Order statistics (Ch. 9)</b> Order statistics (find $i$ -th smallest element); Randomized selection, deterministic selection in linear time <i>Homework 5 due</i>
Th 3/5	<b>Red-black trees (Ch. 13.1, 13.2, 13.3)</b> Red-black tree property, rotations, insertion; abstract data types, ADT dictionary

Date	Material
Tu 3/10	<b>SPRING BREAK</b>
Th 3/12	<b>SPRING BREAK</b>
Tu 3/17	<b>Dynamic programming (Ch. 15.4, +)</b> Fibonacci, binomial coefficient, LCS: fill table, then construct solution from the table. <i>Programming project 1 due</i> <i>Homework 6 assigned</i>
Th 3/19	<b>Dynamic programming (Ch. 15.3, 15.4, 16.2, +)</b> 0-1 Knapsack; general outline of dynamic programming: Optimal substructure (recurrence), overlapping subproblems, fill table bottom-up or by memoization.
Mo 3/23	<b>Drop deadline to drop with a 'W'</b>
Tu 3/24	<b>Greedy algorithms (Ch. 16.2, problem 16-1 on page 402)</b> Greedy algorithms (greedy-choice property, optimal substructure). Making change, fractional knapsack. <i>Homework 6 due; homework 7 assigned</i>
Th 3/26	<b>Elementary Graph Algorithms (Ch. 22.1–22.2)</b> Representations of graphs, breadth-first search (BFS) <i>Programming project 2 assigned</i>
Tu 3/31	<b>Elementary Graph Algorithms (Ch. 22.3–22.4)</b> Depth-first search (DFS), topological sort <i>Homework 7 due</i>
Th 4/2	<b>Test 2</b> Material from 2/12 until 3/24 (inclusive)
Tu 4/7	<b>Minimum Spanning Trees (Ch. 23)</b> Prim (grows single tree), Kruskal (grows forest; uses union/find data structure) <i>Homework 8 assigned</i>
Th 4/9	<b>Single-source shortest paths (Ch. 24 without 24.4)</b> 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 4/14	<b>All-Pairs Shortest Paths (Ch. 25.2)</b> Dynamic programming: Floyd-Warshall <i>Homework 8 due; homework 9 assigned</i>
Th 4/16	<b>P and NP (Ch. 34)</b> Decision problems, definition of classes P and NP, polynomial-time reductions
Tu 4/21	<b>P and NP (Ch. 34)</b> NP-hardness, NP-completeness; Show that problems are NP-complete by reducing from other problems <i>Homework 9 due; homework 10 assigned</i>
Th 4/23	<b>P and NP (Ch. 34)</b> TSP, Clique, Independent Set, Vertex Cover, Hamilton Path, Hamilton Circuit
Tu 4/28	<b>Review for Final Exam</b> Review for final exam <i>Homework 10 due; programming project 2 due</i>

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

**The comprehensive final exam will be on Monday May 4th, 10:30am – 1pm.**