

**CS 3510 A: Design and Analysis of Algorithms**  
**Fall 2008**  
**So far in the course**

**Aug 18:** Introduction to the course; Models of computation.

**Aug 20:** Exponential versus polynomial time algorithms: Fibonacci numbers (Chapter 0); Divide and Conquer: Integer multiplication (Chapter 2).

**Aug 22:** Integer multiplication continued (Chapter 2). Home work 1 handed out.

**Aug 25:** Merge sort (Chapter 2). Counting inversions.

**Aug 27:** Matrix multiplication: Strassen's algorithm. Divide and conquer recurrences. (Chapter 2 and notes).

**Aug 29:** Divide and conquer recurrences continued. A lower bound for sorting using comparisons (Chapter 2).

**Sep 1:** Holiday - Labor day.

**Sep 3:** Home work 1 due. Lower bound for sorting using comparisons continued. Median finding. (Chapter 2).

**Sep 5:** Test 1.

**Sep 8:** Lower bound for sorting using comparisons continued. Selecting the  $k$ -th smallest element. (Chapter 2).

**Sep 10:** Fast Fourier Transform. (Section 2.6).

**Sep 12:** Fast Fourier Transform continued. (Section 2.6). Home work 2 handed out.

**Sep 15:** Fast Fourier Transform continued. (Section 2.6).

**Sep 17:** Fast Fourier Transform continued. (Section 2.6).

**Sep 19:** Graph representations (Section 3.1). Depth-first search: basic algorithm; connected components; types of edges (Section 3.2).

**Sep 22:** Depth-first search - linear time algorithms, connected components, types of edges. (Section 3.3).

**Sep 24:** Depth-first search - back edges, cycles, topological sort. (Section 3.3).

**Sep 26:** Test 2

**Sep 29:** Depth-first search - Strongly connected components (Section 3.4).

**Oct 1:** Strongly connected components Example (Section 3.4). Shortest path problems. Breadth-first search. (Sections 4.1, 4.2).

**Oct 3:** Dijkstra's algorithm derivation and example. (Section 4.4). Homework 3 posted.

**Oct 6:** Dijkstra's algorithm continued; priority queues using arrays. (Sections 4.4, 4.5).

**Oct 8:** Priority queue using binary heaps; (Section 4.5). Heap-sort algorithm.

**Oct 10:** Bellman-Ford algorithm; detecting negative cycles; shortest paths in directed acyclic graphs. (Sections 4.6, 4.7).

**Oct 13:** Fall recess.

**Oct 15:** Transitive closure; Floyd-Warshall algorithm. (Section 6.6)

**Oct 17:** Test 3.

**Oct 20:** Minimum spanning trees: The cut property, Prim's algorithm. (Section 5.1)

**Oct 22:** Minimum spanning trees: Kruskal's algorithm, Union-find data structure. (Section 5.1)

**Oct 24:** Huffman encoding. (Section 5.2)

**Oct 27:** Horn Formulas (Section 5.3) Dynamic Programming (Chapter 6)

**Oct 29:** Dynamic programming: chain matrix multiplication (Section 6.5), longest increasing subsequence (Section 6.2).

**Oct 31:** Dynamic programming: Edit distance (Section 6.3). Knapsack (Section 6.4).

**Nov 3:** Maximum flow: definitions, residual networks. (Section 7.2).

**Nov 5:** Maximum flow: residual networks, Ford-Fulkerson algorithm (Section 7.2).

**Nov 10:** Maximum flow: max-flow min-cut theorem. (Section 7.2).

**Nov 12:** Maximum flow: applications; bipartite perfect matching. (Section 7.2).

**Nov 14:** NP-Completeness: Motivation and examples (Section 8).

**Nov 17:** NP-Completeness: Reductions: 3CNFSAT to Independent SAT (Section 8.3).

**Nov 19:** NP-Completeness: CNFSAT to 3CNFSAT (Section 8.3).

**Nov 21:** NP-Completeness: 3CNFSAT to 3coloring, 3CNFSAT to Subset Sum.

**Nov 26:** Unsolvable problems (pages 263-264 of the text).

**Dec 1:** Modular Exponentiation, Euclid's algorithm (section 1.2), generating random primes (section 1.3.1).

**Dec 3:** RSA (section 1.4).

**Dec 5:** Wrap-up.