Engineering and Applied Science Programs for Professionals
Whiting School of Engineering
Johns Hopkins University
605.621 Foundations of Algorithms
Course Outline

This outline provides an overview of the course and assignments by week. Please remember to check the posted assignments, modules and calendar for specific due dates.

Each course module runs for a period of seven (7) days, referred to as one week. Due dates for readings and other assignments are referred to by the day of the module Week in which they are due. For example, if a reading assignment is to be completed by Day 3 and the module started on Tuesday, then the reading assignment should be completed by Thursday or the 3rd day of the module.

In looking at the order of the topics listed below, some may ask, "Why are we doing things in this order? The order does not track with the order of topics in the textbook at all. Is it just random?" In reality, this ordering of topics was chosen very carefully to emphasize a "foundational" approach to understand the design and analysis of algorithms. Unfortunately, while the textbook we chose is most definitely the best one available, neither this book nor any other (that the instructor is aware) organizes the topics this way.

At a macro level, the course is divided into two sections - analysis methods and algorithm design. We begin with analysis so that we can apply the tools learned later when designing algorithms. That way we can assess how good (or bad) our designs are.

Within the analysis section, we follow a careful progression from basic tools to more advanced topics. The basics begin with definitions and a review of data structures. Once we recall the data structures available to us, we are ready to hit the math. First we consider simple algorithm analysis and recurrence relations. This will enable us to consider worst-case analysis. Next we move on to probabilistic methods (for average case analysis) and amortized methods (to tighten our complexity bounds). Finally, we address one of the most important topics in algorithms - complexity. This is where we consider, not just the typical logarithmic, linear, quadratic, and related complexities but the really hard problems too.

With the basic theoretical tools behind us, we move into the second section where we study classes of algorithms and methods for designing them. As with the first half, we start gently by reminding the students of methods for sorting and finding particular elements in order. We group sorting and selection algorithms together because they are complementary problems. From there, we return to some of the data structures roots by considering algorithms that work on graph-based data structures. The final graph-based problem, network flow, sets us up for the next topic where we examine optimization methods. Here we consider two widely used (and extremely powerful) methods - dynamic programming and linear programming. When we get to linear programming, we revisit network flow because we can use linear programming to solve such problems. We conclude the course by returning to the issue of solving really hard problems. Here we consider randomized methods and approximation algorithms. Thus the flow parallels the flow in the first half and wraps the course up in a nice tidy bow.

To sum things up, it is unfortunate the textbook was not organized as described above. Fortunately, most of the chapters are written to be relatively independent, thus permitting us to cover them in this somewhat peculiar order. While the order may be disconcerting at first, I genuinely hope everyone sees the "method" to my "madness" and comes to agree that this was a reasonable approach to covering the field.

Module	Topic	Assignment
1	Introduction and Data Structures	Introduction and Data Structures, HW 1 Assigned, Sections 2.2: Analyzing Algorithms, 2.3.1: Designing Algorithms, 3.1: Asymptotic Notation
2	Data Structures	CA 1 Assigned, Sections 12.2: Querying a Binary Search Tree, 12.3: Insertion and Deletion, 21.1: Disjoint Set Operations, 21.2: Linked-list Representation of Disjoint Sets, 21.3: Disjoint, HW 1 and CA 1 Due on Day 7
3	Basic Analysis	HW 2 and PA 1 Assigned, Sections A.1: Summation Formulas and Properties, A.2: Bounding Summations, 4.3: The Substitution Method for Solving Recurrences, 4.4: The Recursion Tree Method for Solving Recurrences, 4.5: The Master Method for Solving Recurrences
4	Advance Analysis	CA 2 Assigned, Sections 17.1: Aggregate Analysis, 17.2: The Accounting Method, 17.3: The Potential Method, 17.4: Dynamic Tables (When you get to 17.4.2, read the PDF.), HW 2 and CA 2 are due on day 7 of this module.
5	NP-Complete I	HW 3 Assigned, Sections 34.2: Polynomial-Time Verification, 34.3: NP-Completeness and Reducibility, PA 1 Due on Day 7 of this module.
6	NP-Completeness II	CA 3 and PA 2 Assigned, Sections 34.4: NP-Completeness Proofs, 34.5.3: The Hamiltonian-Cycle Problem, 34.5.4: The Traveling-Salesman Problem HW 3 and CA 3 are due on day 7 of this module
7	Sorting	HW 4 Assigned, Chapter 6: Heapsort, Chapter 7: Quicksort, Sections 7.4: Analysis of Quicksort, 8.1:Lower Bounds for Sorting, 8.2: Counting Sort, 8.3: Radix Sort, 8.4: Bucket Sort
8	Selection	CA 4 Assigned, Chapter 9: Medians and Order Statistics, Chapter 14: Augmenting Data Structures, HW 4 and CA 4 are due on day 7 of this module
9	Graph Algorithms I	HW 5 Assigned, Sections 22.2: Breadth-First Search, 22.3: Depth-First Search, 22.5: Strongly Connected Components, 23.1: Growing a Minimum Spanning Tree, 23.2: The Algorithms of Kruskal and Prim, PA 2 Due on Day 7 of this module
10	Graph Algorithms II	CA 5 and PA3 Assigned, Sections 24.3: Dijstra's Algorithm, 24.1: The Bellman-Ford Algorithm, 26.1: Flow Networks, 26.2: The Ford-Fulkerson Method, 26.2: The Ford-Fulkerson Method (The Edmonds-Karp Algorithm), 26.3: Maximum Bipartite Matching, HW 5 and CA 5 are due on day 7 of this module
11	Optimization I	HW 6 Assigned, Sections 16.2: Elements of the Greedy Strategy, 15.2: Matrix Chain Multiplication, 15.3: Elements of Dynamic Programming, 15.4: Longest Common Subsequence, 15.5: Optimal Binary Search Trees.
12	Optimization II	CA 6 Assigned, Sections 29.1: Standard and Slack Forms, 29.2: Formulating Problems as Linear Programs, 29.3: The Simplex Algorithm, 29.4: Duality, HW 6 and CA 6 are due on day 7 of this module
13	Randomized Algorithms	HW 7 Assigned, Sections C.2: Probability, C.3: Discrete Random Variables, C.4: The Geometric and Binomial Distributions, 5.2: Indicator Random Variables, 5.3: Randomized Algorithms, PA 3 due on day 7 of this module
14	Approximation Algorithms	CA 7 Assigned, Sections 35.1: The Vertex-Cover Problem, 35.2: The Traveling Salesman Problem, 35.3: The Set-Covering Problem, 35.4: Randomization and Linear Programming, HW 7 and CA 7 are due on day 7 of this module