Syllabus

Description. An introduction to fundamental data types, algorithms, and data structures. Our emphasis is on applications and scientific performance analysis of Java implementations.

- Part I focuses on elementary data structures, sorting, and searching. Topics include union–find, binary search, stacks, queues, bags, insertion sort, selection sort, shell sort, quicksort, 3-way quicksort, merge sort, heapsort, binary heaps, binary search trees, red–black trees, separate-chaining and linear-probing hash tables, Graham scan, and kd-trees.
- Part II focuses on graph and string-processing algorithms. Topics include depth-first search, breadth-first search, topological sort, Kosaraju–Sharir, Kruskal, Prim, Dijkstra, Bellman–Ford, Ford–Fulkerson, LSD radix sort, MSD radix sort, 3-way radix quicksort, multiway tries, ternary search tries, Knuth–Morris–Pratt, Boyer–Moore, Rabin–Karp, regular expression matching, run-length coding, Huffman coding, LZW compression and Burrows–Wheeler transform. Part II also introduces reductions and intractability, including the P = NP problem.

Prerequisites. The programming prerequisite for the course is familiarity with Java, including loops, arrays, functions, recursion, and objects. We introduce advanced features of Java as necessary (such as generics and iterators). The mathematics prerequisite is high-school algebra.

Lectures. There are two lectures (75 minutes each) per week. Each lecture is broken up into about 4–6 segments, separated by interactive quiz questions to help you process and understand the material.

Readings. Our textbook is the basic reference for the material we will be covering. Although the lectures are designed to be self-contained, we will assign suggested (but optional) readings for students who wish more extensive coverage of the material.

- *Algorithms, 4th Edition* by Robert Sedgewick and Kevin Wayne, Addison-Wesley Professional, 2011, ISBN 0-321-57351-X.

You can purchase the textbook in either hardcover or electronic format from amazon.com.

Our booksite is open to everyone and contains a wealth of supplementary information, including synopses of the textbook and Java code that you will be using throughout the course.

Exercises. (*Sorry, Coursera did not migrate the exercises from the old platform*.) There is one exercise set associated with each lecture. An exercise set consists of three drill questions, designed to ensure that you understand the basics. We expect that each exercise set will take approximate 10–30 minutes to complete. You may attempt each exercise set up to 10 times (your best score is recorded); you will receive a different set of questions on each attempt. Solutions are provided after each attempt.

Programming assignments. There is one Java programming assignment per week. Some of the programming assignments require that you *implement* a data structure or algorithm from scratch; others require that you *apply* a data structure or algorithm to solve a problem from science, commerce, or recreation. We expect that each programming assignment will take approximately 4–20 hours to complete, depending on your programming experience and how far beyond the basic performance requirements you wish to go. You may attempt each programming assignment up to 10 times (your best score is recorded). After each attempt, we will provide feedback on correctness and efficiency.

Java. The assignments require that you use the Java programming language. More specifically, the auto grader uses Java 8 to compile, execute, and test your Java programs. We recommend our custom IntelliJ-based programming environment for Mac OS X, Windows, or Linux, but you are free to use any IDE. More details about the auto grader are available in the Assessment Guide.

Final exam. (*Sorry, Coursera did not migrate the final exam from the old platform*.) The final exam is cumulative and designed to make sure you understand how each algorithm works and when it is effective. The final does not involve Java programming. Many of the questions will be based on questions from the exercises and in-lecture quizzes. You may attempt the final exam up to 3 times (your best score is recorded).

Job interview questions. Each week, we will also provide a few algorithmic job interview questions based on the material for the week, inspired by questions asked at leading technology companies. The questions are for self-enrichment and are not assessed; however, we will provide some hints and you are welcome to discuss solutions in the discussion forums with your classmates.

Discussion forums. Please be sure to make use of the discussion forums if you need help, and please contribute to the forums if you think that you can provide help. Such forums have proven to be an important and fun part of the online course experience.

Honor code. All students in the course must agree to abide by the Coursera honor code. In particular, do *not* post solutions or partial solutions to programming assignments (including in the Discussion Forums or public code repositories such as GitHub); however, you are

permitted to discuss general ideas and problem-solving approaches. You are also permitted to discuss solutions to exercises and job interview questions.

Certificates. No certificates, statements of accomplishment, or other credentials will be awarded in connection with this course.

Credits. We are grateful to many for help in developing this course. We thank Daniel Kearns, Laura Shaddock, and Jeffrey Himpele for video production; Josh Hug for programming assignment infrastructure; Andrew Morrison for exercise infrastructure; and Jack Magill for migrating the programming assignment infrastructure to the new Coursera platform. We also thank Daphne Koller for inspiring us to teach a MOOC.