WELCOME: Welcome to Part 3 of the Algorithms Specialization: Greedy Algorithms, Minimum Spanning Trees, and Dynamic Programming! Like the previous two parts, the course will have four weeks of lectures and assignments, followed by a final exam. Here are the highlights of the course's first week.

TWO MOTIVATING APPLICATIONS: We begin with a fairly non-technical discussion of two motivating applications --- distributed shortest-path routing in the Internet, and sequence alignment --- to build excitement for the tools that you'll acquire later in this course (and also in Part 4 of the specialization).

INTRODUCTION TO GREEDY ALGORITHMS: The focus of this week and the next is the greedy algorithm design paradigm. These two non-technical videos discuss the pros and cons of this paradigm and describe a cool application to the optimal management of the contents of a cache.

A SCHEDULING APPLICATION: Scheduling problems come up all the time (e.g., how should a shared resource be allocated?) and greedy algorithms are often useful for them. We'll discuss a specific success story --- minimizing the weighted sum of completion times of a bunch of tasks --- in detail. The correctness proof furnishes a particularly clean example of an "exchange argument."

PRIM'S MST ALGORITHM: The minimum spanning tree (MST) problem, in addition to enjoying several applications, is a uniquely great problem for the study of greedy algorithms. Unusually, several different greedy algorithms always compute an optimal solution. We begin here with the Dijkstra-esque Prim's algorithm. The correctness proof requires understanding the subtly beautiful structure of cuts in graphs, while its blazingly fast implementation relies on a deft application of the heap data structure.

VIDEOS AND SLIDES: A reminder that videos can be streamed or downloaded and watched offline (recommended for commutes, etc.). We are also providing PDF lecture slides (typed versions of what's written in the lecture videos), as well as subtitle files (in English and in some cases other languages as well). And if you find yourself wishing that I spoke more quickly or more slowly, note that you can adjust the video speed to accommodate your preferred pace.

HOMEWORK #1: The first problem set consists of 5 problems, about greedy scheduling algorithms and minimum spanning trees. The first programming assignment asks you to implement some of the algorithms that we've covered, run them on large inputs, and enter the answer. For the seasoned programmers out there looking for an additional challenge, try doing the programming assignments in a programming language that you don't already know!

DISCUSSION FORUMS: Discussion forums play an absolutely crucial role in massive online courses. If you have trouble understanding a lecture or completing an assignment, you should turn to the forums for help. After you've mastered the lectures and assignments for a given week, I hope you'll contribute to the forums and help out your fellow learners. While I won't have time to carefully monitor the discussion forums, I'll check in and answer questions whenever I find the time.

SUGGESTED READINGS FOR WEEK 1: Abbreviations in suggested readings refer to the following textbooks:

* CLRS - Cormen, Leiserson, Rivest, and Stein, *Introdution to Algorithms (3rd edition)*
* DPV - Dasgupta, Papadimitriou, and Vazirani, *Algorithms*
* KT - Kleinberg and Tardos, *Algorithm Design*
* SW - Sedgewick and Wayne, *Algorithms (4th edition)*
* CLRS: Chapter 16 (Sections 1 and 2) and Chapter 23
* DPV: Sections 5.1.1, 5.1.2, and 5.1.5
* KT: Sections 4.1, 4.2, 4.3, and 4.5
* SW: Section 4.3