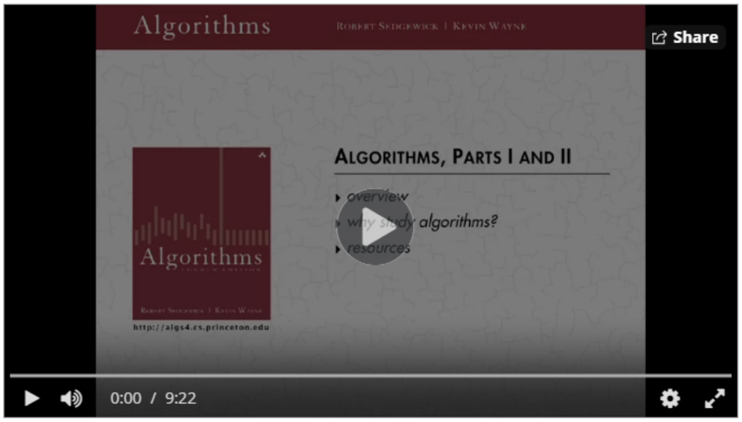
Course Introduction



Save Note Discuss Download Share

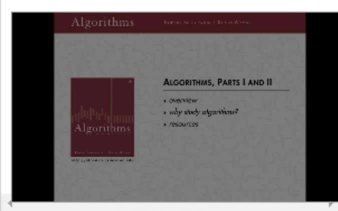
English Help Us Translate

0:01 Welcome. I'm Bob Sedgewick, professor of computer science at Princeton. This is our online course Algorithms developed by myself and Kevin Wayne here at Princeton. We're gonna start with an overview discussion of why you might want to study algorithms and a little bit of discussion about the resources that you need to take this course. So, what is this course? It's an intermediate level survey course on algorithms. We're going to concentrate on programming and problem solving in the context of real applications, and our focus is going to be on two things, Algorithms which are methods for solving problems and data structures which store the information associated in problem, with a problem and go hand in hand with algorithms. These are the basic topics that we'll cover in part one and part two of the course. The first part is data type sorting and searching. We'll consider a number of data structures and algorithms that are basic to all the methods we consider including stacks, queues, bags and priority queues. Then we'll consider classic algorithms for sorting, putting things in order. That's quicksort, mergesort, heapsort and radix sorts. And we'll consider classic methods for searching. Including binary search trees, red-black binary search trees and hash tables. The second part of the course is for more advanced algorithms including graph algorithms, classic graph searching algorithms, minimum spanning tree and shortest path algorithms, algorithms for processing strings including regular expressions and data compression. And then some advanced algorithms that make use of the basic algorithms that we developed earlier in the course. So, why should one study algorithms? Well, their input, impact is very broad and far-reaching. From the internet to biology to, commercial computing, computer graphics, security, multimedia, social networks, and scientific applications, algorithms are all around us. They're used for movies and video games, for particle collision simulation, they're used to study the genome, and all manner of other applications. So, that's one important reason to study algorithms, their impact is broad and far-reaching. Algorithms are also interesting to study, because they, they have ancient roots. Now the first algorithm we studied goes back to 300 B.C., dating at least to Euclid. The concept of an algorithm was formalized actually here at Princeton, by Church and Turing, in the 1930s. But most algorithms that we consider, were discovered in recent decades. In fact, some were discovered by undergraduates in a course, course like this. And there's plenty of other algorithms waiting to be discovered by students like you. The main reason that people study algorithms, is to be able to solve problems that it could not otherwise be addressed. For example, in the first lecture, we're going to talk about the network connectivity problem, where the problem is, given a large set of items that are connected together pairwise is there a way to get from one to another with a path through the connections. As you can see from this example, it's not clear whether or not there's such a path, we need a computer program to do it, in fact, we need an efficient algorithm to do it. In this case the answer is that there is such a path. Another reason to study algorithms is for intellectual stimulation. Algorithms are very interesting objects to study. Don Knuth who wrote several books on, on algorithms and was a pioneer in the field said that, "An algorithm must be seen to be believed." You can't just think about an algorithm you have to work with it. Another quote from Francis Sullivan, says, "The great algorithms are the poetry of computation." Just like verse, they can be terse, elusive, dense, and even mysterious. But once unlocked, they cast a brilliant new light on some aspect of computing.

Click the "Save Note" button when you want to capture a screen. You can also highlight and save lines from the transcript below. Add your own notes to anything you've captured.

Click the "Save Note" button when you want to capture a screen. You can also highlight and save lines from the transcript below. Add your own notes to anything you've captured.

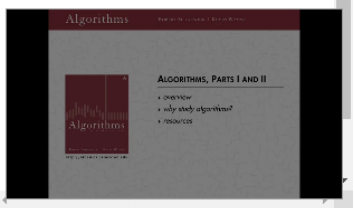
Click the "Save Note" button when you want to capture a screen. You can also highlight and save lines from the transcript below. Add your own notes to anything you've captured.



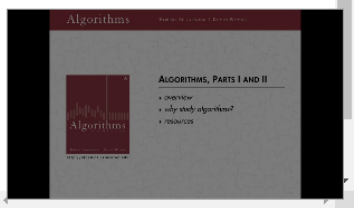
Click the "Save Note" button when you want to capture a screen. You can also highlight and save lines from the transcript below. Add your own notes to anything you've captured.

Algorithms are interesting for intellectual stimulation. Another reason many people study algorithms and I suspect many of you, is it's necessary to understand good algorithms, efficient algorithms, a good data structures in order to be a proficient programmer. Linus Torvalds, who created lin, Linux, says that the difference between a bad programmer and a good one is whether he considers his code or his data structures more important. Bad programmers worry about the code, good programmers worry about data structures, and their relationships. And, I might add, the algorithms that process them. Niklaus Wirth, another pioneer in computer science, wrote a famous book called Algorithms + Data Structures = Programs. [cough]. Another reason nowadays to study algorithms is that, they have become a common language for understanding, nature. Algorithms are computational models, and algorithmic models are replacing mathematical models in scientific inquiry. In the twentieth century, math, scientists developed mathematical models to try to understand natural phenomenon. It soon became clear that those mathematical models were difficult to solve. It was difficult to create solutions, to be able to test hypotheses against natural phenomenon. So, more and more and more now a days people are developing computational models, where they attempt to simulate what might be happening in nature in order to try to better understand it. Algorithms play an extremely important role in this process. And we'll see some examples of this in this course. Another important reason is that if you know effect, how to effectively use algorithms and data structures you're going to have a much better chance at interviewing for a job in the technology industry then if you don't. So, here's a bunch of reasons that I just went through for studying algorithms. Their impact's broad and far-reaching, they have old roots and present new opportunities, they allow us to solve problems that could not otherwise be addressed, you can use them for intellectual stimulation to become a proficient programmer. They might unlock the secrets of life in the universe, and they're good for fun and profit. In fact, a pr ogrammer might ask, why study anything else? Well, there's plenty of good reasons to study other things, but I'll submit there's no good reason not to study algorithims. [cough] So, for this course we have two resources that I want to talk about and make sure that people are familiar with before entering into the content. This is a publishing model that Kevin Wayne and I developed and have been using for many years, and we think it's a very effective way to support the, kinds of lectures that we're going to be giving in this course. Down at the bottom, and it's optional for this course, we have a text book. It's a traditional, text book that extensively covers the topics in the course, in fact many more topics than we can present in lecture. And then supporting that textbook, is free online material that we call the book site. You can go to books, the book site to see the lecture slides. But more important, there's code, there's exercises, tere's a great deal of information there. In fact, maybe ten times what's in the book, including a summary of the content. So, during this course you'll be referring to the book site frequently while working online. People often ask about prerequisites. We're assuming that people who take this course know how to program, and know the basics of loops, arrays, functions. They have some exposure to object oriented programming and recursion. We use the Java language, but we don't dwell on details of Java, we mostly use it as an expository language. We do some math, but not advanced math. If you want to review the material that we think is prerequisite for the material in this course, you can do a quick review by looking at sections 1.1 and 1.2 of the book. Either at the book site or in the text book. If you want an in depth review, we have a full text book called, An Introduction to Programming in Java: An Interdisciplinary Approach. There is a book site and text book as well. But, the bottom line is, you should be able t o program, and the quick exercise to get ready is, to write a java program on your computer perhaps using a programming model, as described on the book site. We will provide much more detail information on that as we get into the assignments. You can use your own programming environment if your comfortable with one or you download ours. We have instructions on the web on how to do that.

Click the “Save Note” button when you want to capture a screen. You can also highlight and save lines from the transcript below. Add your own notes to anything you’ve captured.



Click the “Save Note” button when you want to capture a screen. You can also highlight and save lines from the transcript below. Add your own notes to anything you’ve captured.



Click the “Save Note” button when you want to capture a screen. You can also highlight and save lines from the transcript below. Add your own notes to anything you’ve captured.

Click the “Save Note” button when you want to capture a screen. You can also highlight and save lines from the transcript below. Add your own notes to anything you’ve captured.