Browse > Computer Science > Algorithms

Algorithms, Part I

★★★★★ 4.9 7,486 ratings | 98%


Kevin Wayne +1 more instructor

Offered By

PRINCETON UNIVERSITY

Enroll for Free
Starts Aug 24

698,042 already enrolled

About Instructors Syllabus Reviews Enrollment Options FAQ

About this Course

997,806 recent views

This course covers the essential information that every serious programmer needs to know about algorithms and data structures, with emphasis on applications and scientific performance analysis of Java implementations. Part I covers elementary data structures, sorting, and searching algorithms. Part II focuses on graph- and string-processing algorithms.

All the features of this course are available for free. It does not offer a certificate upon completion.

SKILLS YOU WILL GAIN

Data Structure | Algorithms | Java Programming

Learner Career Outcomes

32% started a new career after completing these courses

34% got a tangible career benefit from this course

17% got a pay increase or promotion

100% online
Start instantly and learn at your own schedule.

Flexible deadlines
Reset deadlines in accordance to your schedule.

Intermediate Level

Approx. 53 hours to complete

English
Subtitles: English, Korean, Russian

Instructors

Instructor rating 4.79/5 (569 Ratings)

Kevin Wayne
Phillip Y. Goldman '86 Senior Lecturer
Computer Science
783,995 Learners
4 Courses

Robert Sedgewick
William O. Baker *39 Professor of Computer Science
Computer Science
804,237 Learners
6 Courses

Offered by

Princeton University

Princeton University is a private research university located in Princeton, New Jersey, United States. It is one of the eight universities of the Ivy League, and one of the nine Colonial Colleges founded before the American Revolution.

Coursera's rigorous assignments and broad range of subjects encourage me to keep up with my courses. The quality of the teachers keeps me coming back.

— Sandra O.

People interested in this course also viewed

Algorithms, Part II
Princeton University
1 COURSE

Analysis of Algorithms
Princeton University
1 COURSE

Computer Science: Algorithms, Theory, and Machines
Princeton University
1 COURSE

Computer Science: Programming with a Purpose
Princeton University
1 COURSE

Syllabus - What you will learn from this course

Content Rating 98% (52,206 ratings)

WEEK 1

10 minutes to complete

Course Introduction

Welcome to Algorithms, Part I.

1 video (Total 9 min), 2 readings SEE ALL

9 hours to complete

Union–Find

We illustrate our basic approach to developing and analyzing algorithms by considering the dynamic connectivity problem. We introduce the union–find data type and consider several implementations (quick find, quick union, weighted quick union, and weighted quick union with path compression). Finally, we apply the union–find data type to the percolation problem from physical chemistry.

compression). Finally, we apply the union-find data type to the percolation problem from physical chemistry.

5 videos (Total 51 min), 2 readings, 2 quizzes SEE ALL

1 hour to complete

Analysis of Algorithms

The basis of our approach for analyzing the performance of algorithms is the scientific method. We begin by performing computational experiments to measure the running times of our programs. We use these measurements to develop hypotheses about performance. Next, we create mathematical models to explain their behavior. Finally, we consider analyzing the memory usage of our Java programs.

6 videos (Total 66 min), 1 reading, 1 quiz SEE ALL

WEEK 2

9 hours to complete

Stacks and Queues

We consider two fundamental data types for storing collections of objects: the stack and the queue. We implement each using either a singly-linked list or a resizing array. We introduce two advanced Java features—generics and iterators—that simplify client code. Finally, we consider various applications of stacks and queues ranging from parsing arithmetic expressions to simulating queueing systems.

6 videos (Total 61 min), 2 readings, 2 quizzes SEE ALL

1 hour to complete

Elementary Sorts

We introduce the sorting problem and Java's Comparable interface. We study two elementary sorting methods (selection sort and insertion sort) and a variation of one of them (shellsort). We also consider two algorithms for uniformly shuffling an array. We conclude with an application of sorting to computing the convex hull via the Graham scan algorithm.

6 videos (Total 63 min), 1 reading, 1 quiz SEE ALL

WEEK 3

9 hours to complete

Mergesort

We study the mergesort algorithm and show that it guarantees to sort any array of n items with at most n lg n compares. We also consider a nonrecursive, bottom-up version. We prove that any compare-based sorting algorithm must make at least n lg n compares in the worst case. We discuss using different orderings for the objects that we are sorting and the related concept of stability.

5 videos (Total 49 min), 2 readings, 2 quizzes SEE ALL

1 hour to complete

Quicksort

We introduce and implement the randomized quicksort algorithm and analyze its performance. We also consider randomized quickselect, a quicksort variant which finds the kth smallest item in linear time. Finally, we consider 3-way quicksort, a variant of quicksort that works especially well in the presence of duplicate keys.

4 videos (Total 50 min), 1 reading, 1 quiz SEE ALL

WEEK

4

9 hours to complete

Priority Queues

We introduce the priority queue data type and an efficient implementation using the binary heap data structure. This implementation also leads to an efficient sorting algorithm known as heapsort. We conclude with an applications of priority queues where we simulate the motion of n particles subject to the laws of elastic collision.

4 videos (Total 74 min), 2 readings, 2 quizzes SEE ALL

1 hour to complete

Elementary Symbol Tables

We define an API for symbol tables (also known as associative arrays, maps, or dictionaries) and describe two elementary implementations using a sorted array (binary search) and an unordered list (sequential search). When the keys are Comparable, we define an extended API that includes the additional methods min, max floor, ceiling, rank, and select. To develop an efficient implementation of this API, we study the binary search tree data structure and analyze its performance.

6 videos (Total 77 min), 1 reading, 1 quiz SEE ALL

Show More

Reviews

4.9 ★★★★★ 1561 reviews

5 stars	91.86%
4 stars	6.85%
3 stars	0.74%
2 stars	0.17%
1 star	0.36%

TOP REVIEWS FROM ALGORITHMS, PART I

★★★★★

by RM Jun 1, 2017

This is a great class. I learned / re-learned a ton. The assignments were challenge and left a definite feel of accomplishment. The programming environment and automated grading system were excellent.

★★★★☆

by RB Jun 1, 2020

The course focuses on Java specific implementations of algorithms, so if you're not from a Java background, you would have a little difficulty with the assignments.The course content however is great.

★★★★★

by BJ Jun 3, 2018

Good contents and the logic of the whole course structure is very clear for a novice like me. The weekly homework is also awesome. Would recommend to anyone who wants to learn about computer science.

★★★★★

by RP Jun 11, 2017

Incredible learning experience. Every programmer in industry should take this course if only to dispel the idea that with the advent of cloud computing exponential algorithms can still ruin your day!

View all reviews

Start Learning Today

Enroll for Free

Starts Aug 24

698,042 already enrolled

Frequently Asked Questions

- When will I have access to the lectures and assignments?
- Do I need to pay for this course?
- Can I earn a certificate in this course?
- I have no familiarity with Java programming. Can I still take this course?
- Which algorithms and data structures are covered in this course?
- Which kinds of assessments are available in this course?
- I am/was not a Computer Science major. Is this course for me?
- How does this course differ from Design and Analysis of Algorithms?

More questions? Visit the **Learner Help Center**.