



Analytic Combinatorics for the masses

Robert Sedgewick
Princeton University

This talk is dedicated to the memory of Philippe Flajolet



Philippe Flajolet 1948–2011

Starting point

1975: What are the algorithms that *everyone* should know?

“Everyone” means “everyone using a computer”

- scientists
- engineers
- mathematicians
- software/hardware designers
- cryptanalysts
- COBOL programmers
- . . .



Context

- <1 computer per university, on average
- <10 CS departments worldwide

1981: First edition of *Algorithms* is published



Disseminating knowledge: RS context

Algorithms

Analytic Combinatorics

Introduction to Computer Science

1980

1990

2000

2010

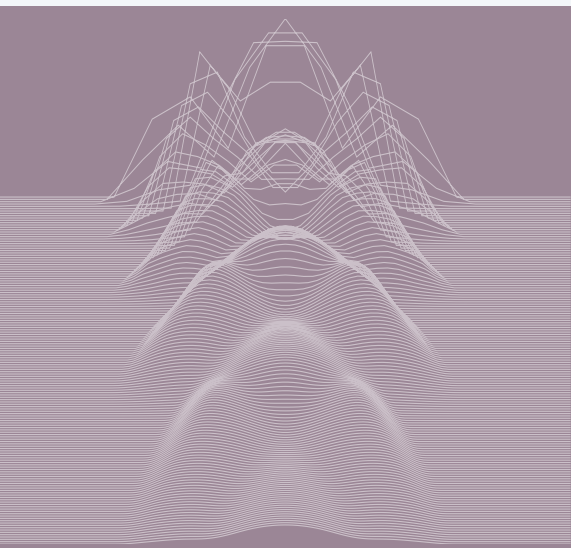
The diagram illustrates the dissemination of knowledge in computer science through textbooks over time. It is organized into three columns: **Algorithms**, **Analytic Combinatorics**, and **Introduction to Computer Science**. A vertical timeline on the left marks the years 1980, 1990, 2000, and 2010.

Algorithms: Shows the evolution of the 'Algorithms' series by Robert Sedgewick. Key titles include 'Algorithms in C', 'Algorithms in C++', and 'Algorithms in Java'. A screenshot of the 'Algorithms, 4th Edition' website is shown at the bottom left, detailing the book's structure and features.

Analytic Combinatorics: Shows the book 'Analytic Combinatorics' by Philippe Flajolet and Robert Sedgewick. A screenshot of the book's cover is shown at the bottom right.

Introduction to Computer Science: Shows the book 'Introduction to Programming in Java: An Interdisciplinary Approach' by Addison-Wesley. A screenshot of the book's website is shown at the bottom right, highlighting its interdisciplinary approach.

Video Player: A video player at the bottom shows a lecture on 'Analysis of mergesort'. The whiteboard displays the recurrence relation $C_n = n \lg(N)$ and the number of comparisons $Q = 2N \ln(N - 2) - (N - 1) \lg(N)$.



Analytic Combinatorics for the masses

- Frequently asked questions
- Disruptive changes
- A way forward
- Analytic combinatorics
- Questions answered



Frequently asked questions

Q. What does MOOC mean?

A. *Massive Open Online Course.*

- *Free*
- Designed for large numbers of "students"

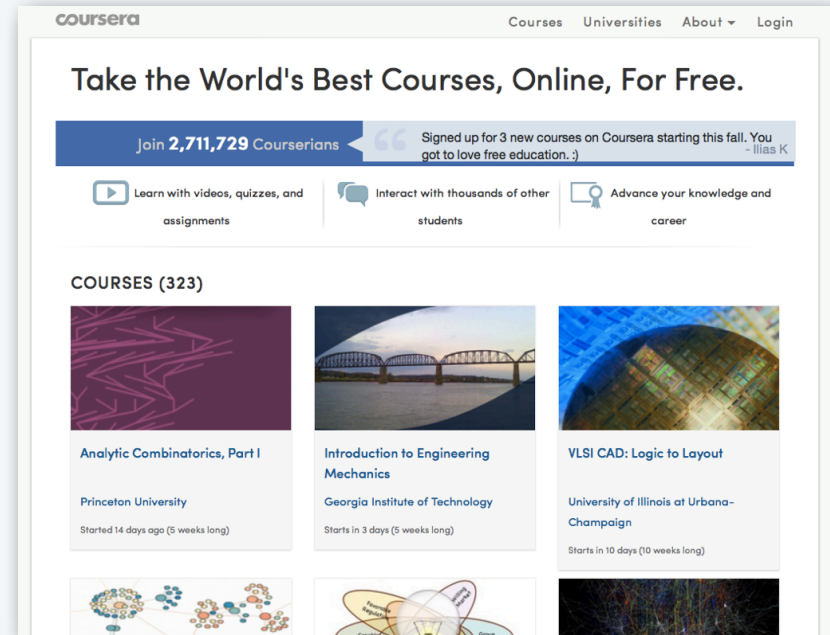
Q. What is the business model?

Q. How are our students getting their money's worth if it's free?

Q. Can we afford to participate?

Q. Will this detract from our "brand"?

Q. Can I get credit for this course?



Just one platform (2013):

300+ courses

2.7 million students

the wrong questions

Note: A more disruptive "connectivist" MOOC concept is widely advocated:

- Crowdsourced decisions and assessment
- Aggregation, remixing, and passing forward content

The right questions

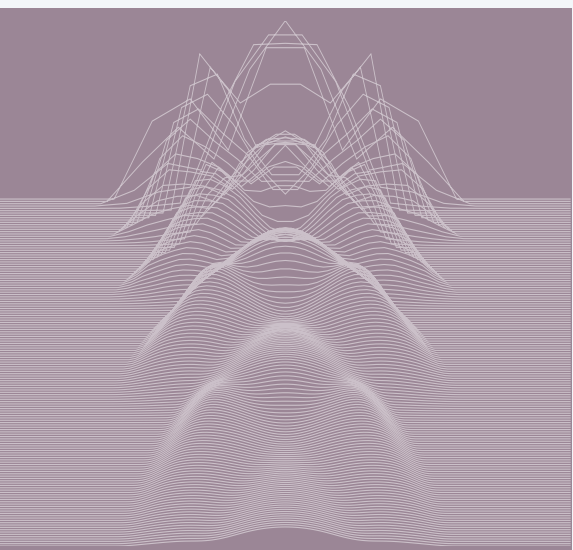
Q. How are we going to disseminate knowledge in the future?

Q. Are universities going to take leadership in helping to decide this question?

Q. If we *can* disseminate knowledge for free, isn't it our responsibility to do so?

Q. What is the purpose of a university?





Analytic Combinatorics for the masses

- Frequently asked questions
- **Disruptive changes**
- A way forward
- Analytic combinatorics
- Questions answered

Seismic changes are afoot

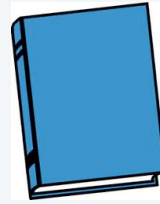
Records?



Newspapers?



Books?



Libraries?



Universities?

*For a millennium, universities have been considered the main societal hub for knowledge and learning. And for a millennium, the basic structures of how universities produce and disseminate knowledge and evaluate students have survived intact...Today, though, the **business of higher education** seems to some as susceptible to tech disruption as other information-centric industries*

The Future of Higher Education - Pew Internet & American Life Project, 2012

Business of higher education ??

A road to ruin.

Grafton: Save the libraries

FUTURE READING

Digitization and its discontents.

by Anthony Grafton

The New Yorker

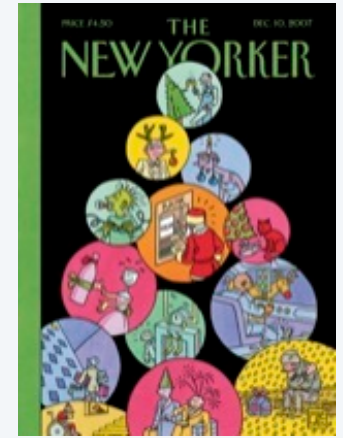
November 5, 2007



*...Sit in your local coffee shop, and your laptop can tell you a lot. If you want deeper, more local knowledge, you will have to take the narrower path that leads between the lions and up the stairs. There—as in great libraries around the world—you’ll use all the new sources, the library’s and those it buys from others, all the time. You’ll check musicians’ names and dates at Grove Music Online, read Marlowe’s “Doctor Faustus” on Early English Books Online, or decipher Civil War documents on Valley of the Shadow. But these streams of data, rich as they are, will illuminate, rather than eliminate, books and prints and manuscripts that only the library can put in front of you. **The narrow path still leads, as it must, to crowded public rooms where the sunlight gleams on varnished tables, and knowledge is embodied in millions of dusty, crumbling, smelly, irreplaceable documents and books.***

RS: Think about the future

The New Yorker
Letter to the editor
Robert Sedgewick
December 10, 2007

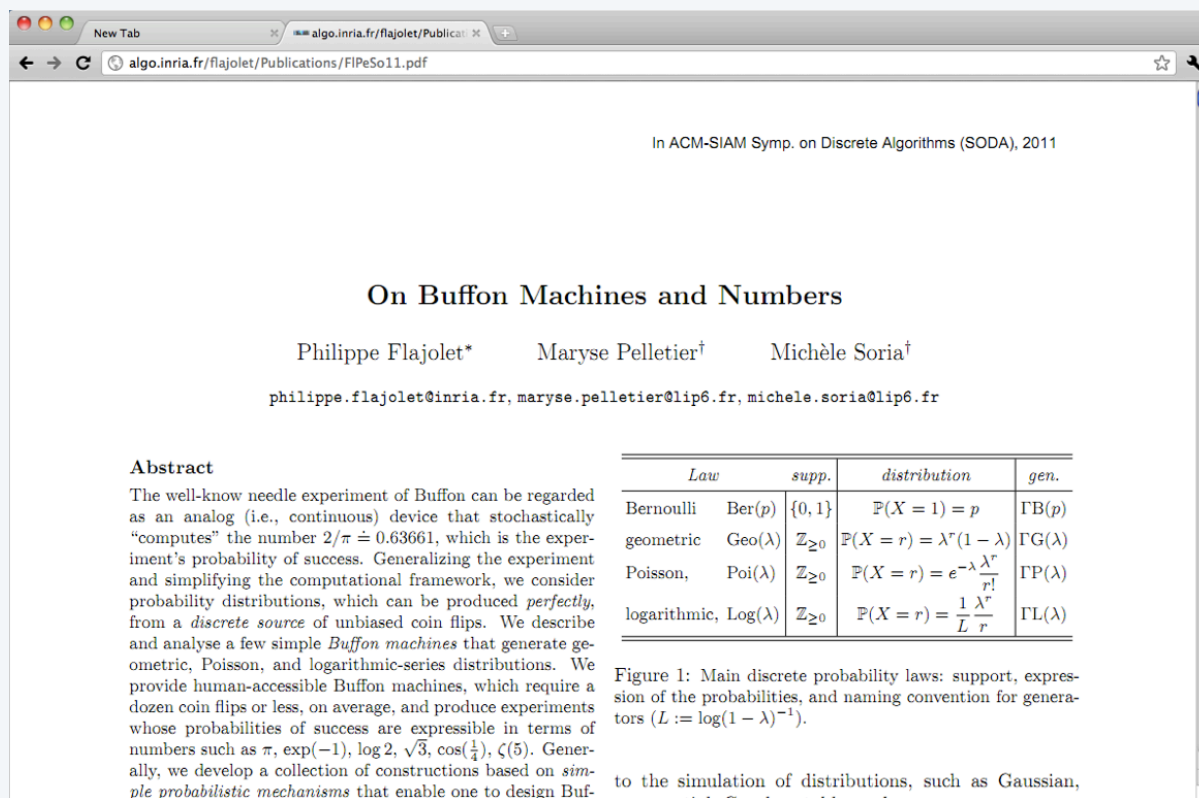


*While Grafton's reservations about putting knowledge online are well taken, I would also point out that there is quite a bit going on now in the academic world that doesn't have much to do with old books. Indeed, as the author of many books, I wonder whether perhaps the book is not quite sacred as a means of disseminating knowledge. **What is the most effective way to produce and disseminate knowledge with today's technology? How can we best structure what we know and learn so that students, researchers, and scholars of the future can best understand the work of today's researchers and scholars?** I think that questions like these are more important and more difficult to address than whether we can put the contents of libraries on the Web.*

Disseminating knowledge I: research papers

When is the last time you visited a library to find a paper?

Did you print the papers to read the last time you refereed a conference?



- Color?
- Links to references?
- Links to detailed proofs?
- Simulations?

↑
why not?

*"I could read it on my iPad
...if I had an iPad"*
D. E. Knuth (2011)

↑
why?

Question: If it will not be read on paper, why write it as if it will?

Prediction: Someone will soon invent the future (should be easy)

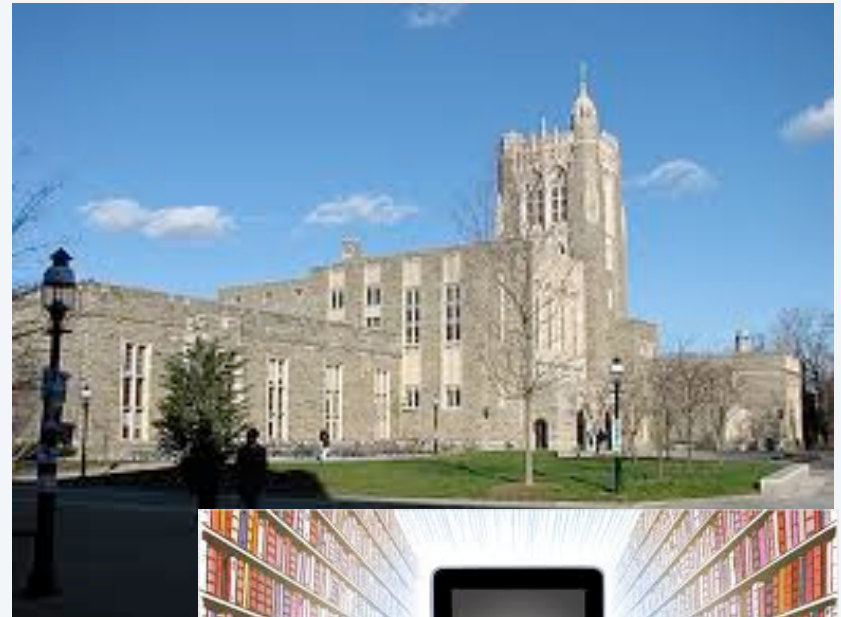
Future of libraries?

1980s

- Students spend significant time in the library
- Faculty members depend on the library for research

2010s

- Students spend significant time online and have *no need* for the library
- Few faculty members in the sciences use the library *at all* for research



2020s?

- A few *book museums* (for Grafton)
- Digital library infrastructure (for everyone else)



How will we disseminate knowledge in the future?

Will universities play a role?

Disseminating knowledge II: Textbooks

We are on a road to ruin

- Prices continue to escalate.
- Students now *rent*, not own books.
- Planned obsolescence? Walled gardens?

\$89.4 million
Rise in dollar sales of
college textbooks,
2004-5

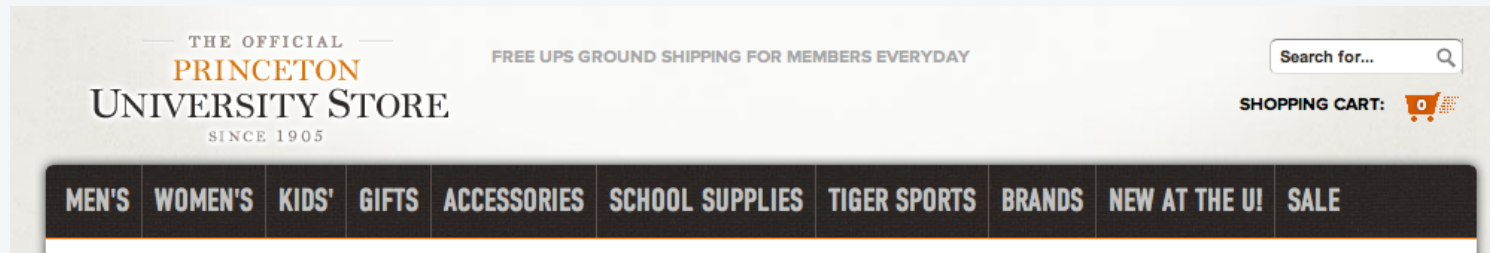
600,000
Decline in number of
new college textbooks
sold, 2004-5



Princeton
U-store
1950s

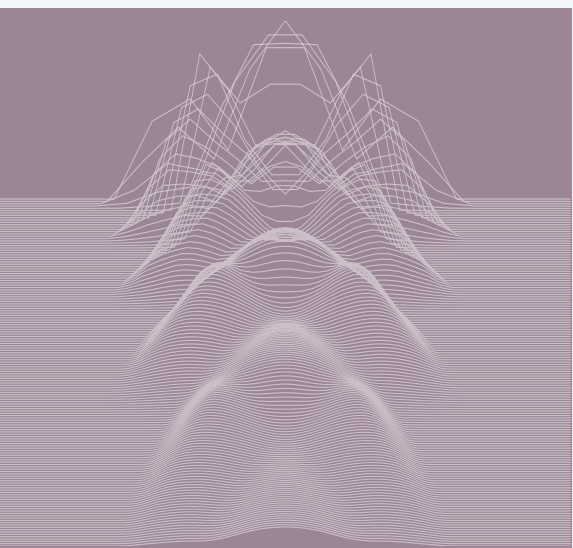


Princeton
U-store
2010s



Is there room for a good textbook?
Will free web resources prevail?

No books!



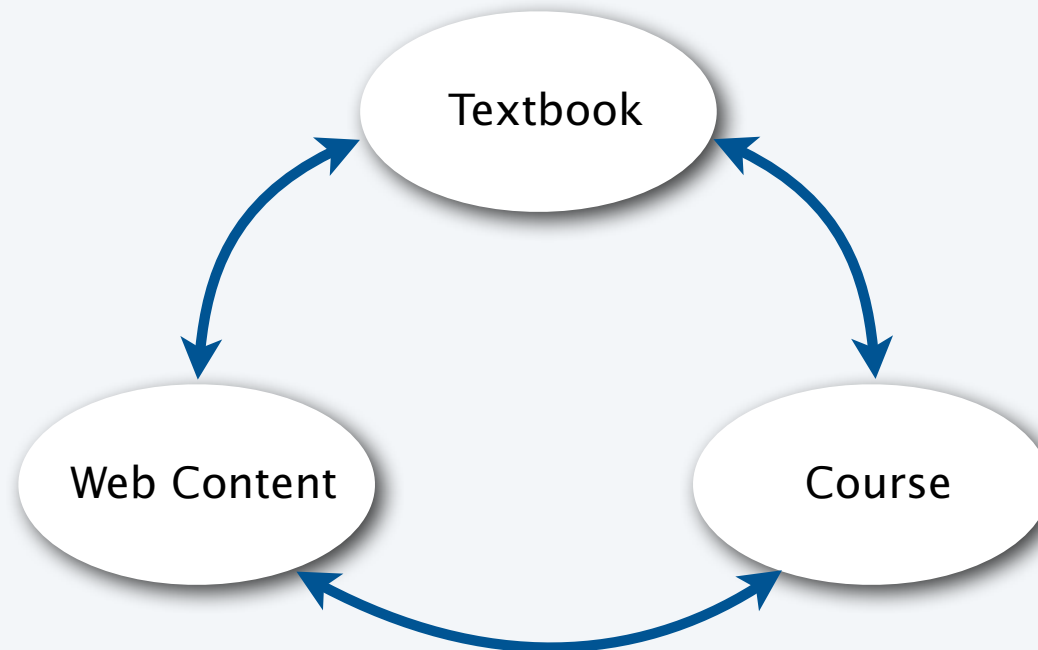
Analytic Combinatorics for the masses

- Frequently asked questions
- Disruptive changes
- **A way forward**
- Analytic combinatorics
- Questions answered

A way forward

embraces technology to integrate three abstractions that are *here to stay*:

1. A **textbook** for use by students to *learn and study* the details of a subject.
2. A **course** that encourages a community of scholars to learn together.
3. **Web content** for use by students to *explore and interact with* the material.



Examples (stay tuned). CS courses at Princeton implemented by RS and Kevin Wayne

The "course" abstraction

has been an essential part of education for a millenium and is *here to stay*

What is a course?

- *Lectures* to introduce to inspire.
- Assignments
- Exams
- *Precepts* to work in small groups.



U. of Bologna, founded in 1088

Purpose of a course

- Enable a "community of scholars" to teach and learn a subject.
- Serve as a building block in a *curriculum*.

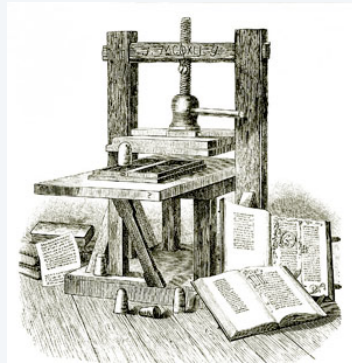
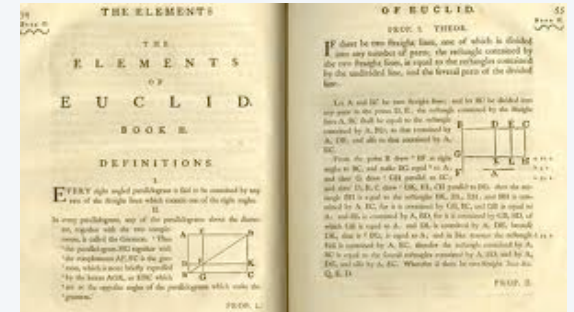


A university lecture in the 1350s

The "textbook" abstraction

has been an essential component in education for centuries and is *here to stay*

Well-understood since the Greeks.



Enabled for the masses by Gutenberg.

Relevant for this talk:



Purpose of a textbook

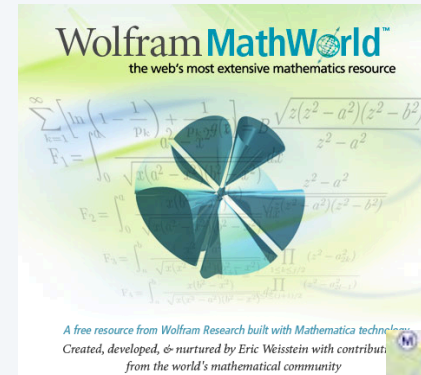
- Articulate what students can reasonably learn about a subject in a semester.
- Provide a reference point for future studies related to the subject.

The "web content" abstraction

is emerging as an essential component in education and is *here to stay*

What is web content?

- Full coverage integrated with web search.
- Always up to date (*dynamic*).
- Content types *not available* in print.



Issues

- Basic properties still evolving.
- Free? Who pays?
- Who creates it? Who maintains it?



WIKIPEDIA

English
The Free Encyclopedia
4 110 000+ articles

Español
La enciclopedia libre
940 000+ artículos

Русский
Свободная энциклопедия
940 000+ статей

Italiano
L'enciclopedia libera
1 000 000+ voci

Português
A enciclopédia livre
780 000+ artigos

日本語
フリー百科事典
835 000+ 記事

Deutsch
Die freie Enzyklopädie
1 510 000+ Artikel

Français
L'encyclopédie libre
1 330 000+ articles

Polski
Wolna encyklopedia
940 000+ hasel

中文
自由的百科全書
610 000+ 條目

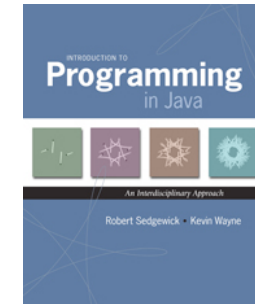


Example 1: Introduction to CS

2000s

- Raw material, digests for "story"
- Enrichment materials
- *Programming model.*

2008



Textbook

Web Content
"Booksite"

Course

1990s

- *Lecture presentations.*
- Assignments
- Exercises and Exams
- *Precepts.*

INTRODUCTION TO PROGRAMMING IN JAVA

a textbook for a first course in computer science
for the next generation
of scientists and engineers

Textbook. Our textbook *Introduction to Programming in Java* [Amazon] takes an interdisciplinary approach to the traditional CS1 curriculum. We treat the elements of programming, using an "objects-in-the-middle" approach to abstraction. A key feature of the book is the manner in which we present the concept by examining its impact on specific applications, taken from materials science to genomics to astrophysics to internet commerce, and around four stages of learning to program:

- *Chapter 1: Elements of Programming* introduces variables; assignment; built-in types of data; conditionals and loops; arrays; and input/output, graphics and sound.
- *Chapter 2: Functions* introduces modular programming. We stress the fundamental concept of dividing a program into components that can be independently debugged, maintained, and reused.
- *Chapter 3: Object-Oriented Programming* introduces data abstraction. We emphasize the concept of a data type and its implementation using Java's class mechanism.
- *Chapter 4: Algorithms and Data Structures* introduces classical algorithms for sorting and searching, and fundamental data structures, including stacks, queues, and symbol tables.

Booksite. Reading a book and surfing the web are two different activities: This booksite is intended for your use while online (for example, while programming and while browsing the web); the textbook is for your use when initially learning new material and when reinforcing your understanding of that material (for example, when reviewing for an exam). The booksite consists of the following elements:

- *Excerpts.* A condensed version of the text narrative for reference while online.
- *Exercises.* Hundreds of exercises and some solutions.
- *Java code.* Hundreds of easily downloadable Java programs and real-world data sets.

To get started. Here are instructions for installing a Java programming environment [Mac OS X · Windows · Linux]. We also provide I/O libraries for reading and writing text and binary data, drawing graphics, and producing sound.

To adopt. Here are some of the [distinctive features](#). To preview our material, you can download the [preface](#). For adoption, please fill out [this form](#) to request a copy of information.

Last modified on January 21, 2013.
Copyright © 2002–2012 Robert Sedgewick and Kevin Wayne. All rights reserved.

10000+ files

2000+ Java programs

50+ animated demos

1.2 million unique visitors in 2011

Example 2: Algorithms, 4th edition

RS: Hey, maybe this could work for "Algorithms" !

KW: No problem, but...it might take some time.

2011

- *Code and data repository.*
- *Enrichment materials*
- *Course materials.*

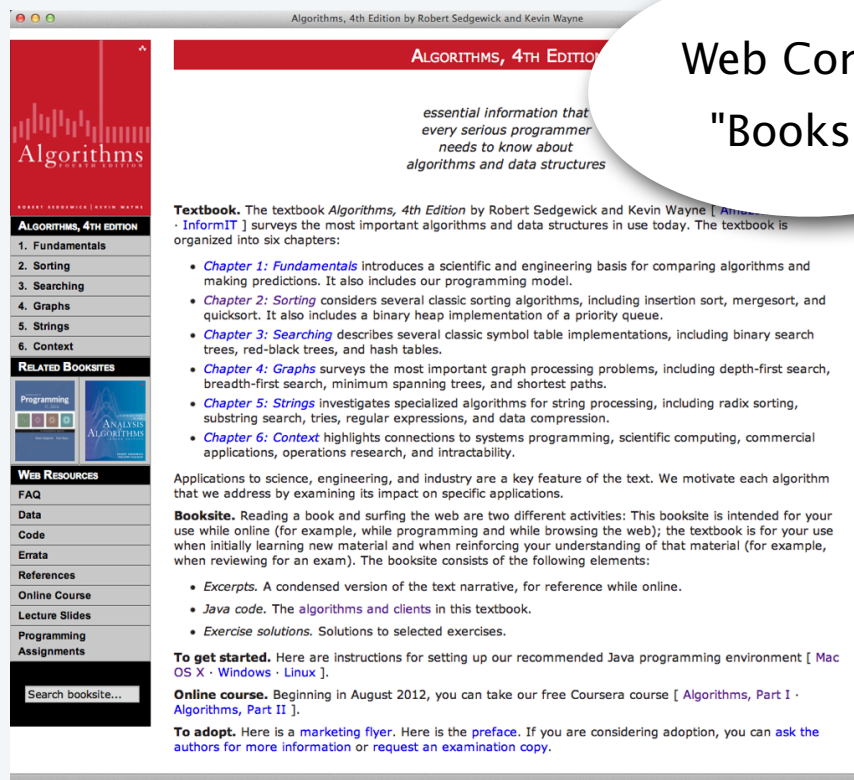
2011



Textbook

Web Content
"Booksite"

Course



ALGORITHMS, 4TH EDITION

essential information that every serious programmer needs to know about algorithms and data structures

Textbook. The textbook *Algorithms, 4th Edition* by Robert Sedgwick and Kevin Wayne [[Amazon](#) · [InformIT](#)] surveys the most important algorithms and data structures in use today. The textbook is organized into six chapters:

- **Chapter 1: Fundamentals** introduces a scientific and engineering basis for comparing algorithms and making predictions. It also includes our programming model.
- **Chapter 2: Sorting** considers several classic sorting algorithms, including insertion sort, mergesort, and quicksort. It also includes a binary heap implementation of a priority queue.
- **Chapter 3: Searching** describes several classic symbol table implementations, including binary search trees, red-black trees, and hash tables.
- **Chapter 4: Graphs** surveys the most important graph processing problems, including depth-first search, breadth-first search, minimum spanning trees, and shortest paths.
- **Chapter 5: Strings** investigates specialized algorithms for string processing, including radix sorting, substring search, tries, regular expressions, and data compression.
- **Chapter 6: Context** highlights connections to systems programming, scientific computing, commercial applications, operations research, and intractability.

Applications to science, engineering, and industry are a key feature of the text. We motivate each algorithm that we address by examining its impact on specific applications.

Booksite. Reading a book and surfing the web are two different activities: This booksite is intended for your use while online (for example, while programming and while browsing the web); the textbook is for your use when initially learning new material and when reinforcing your understanding of that material (for example, when reviewing for an exam). The booksite consists of the following elements:

- *Excerpts.* A condensed version of the text narrative, for reference while online.
- *Java code.* The algorithms and clients in this textbook.
- *Exercise solutions.* Solutions to selected exercises.

To get started. Here are instructions for setting up our recommended Java programming environment [[Mac OS X](#) · [Windows](#) · [Linux](#)].

Online course. Beginning in August 2012, you can take our free Coursera course [[Algorithms, Part I - Algorithms, Part II](#)].

To adopt. Here is a [marketing flyer](#). Here is the [preface](#). If you are considering adoption, you can [ask the authors for more information](#) or [request an examination copy](#).

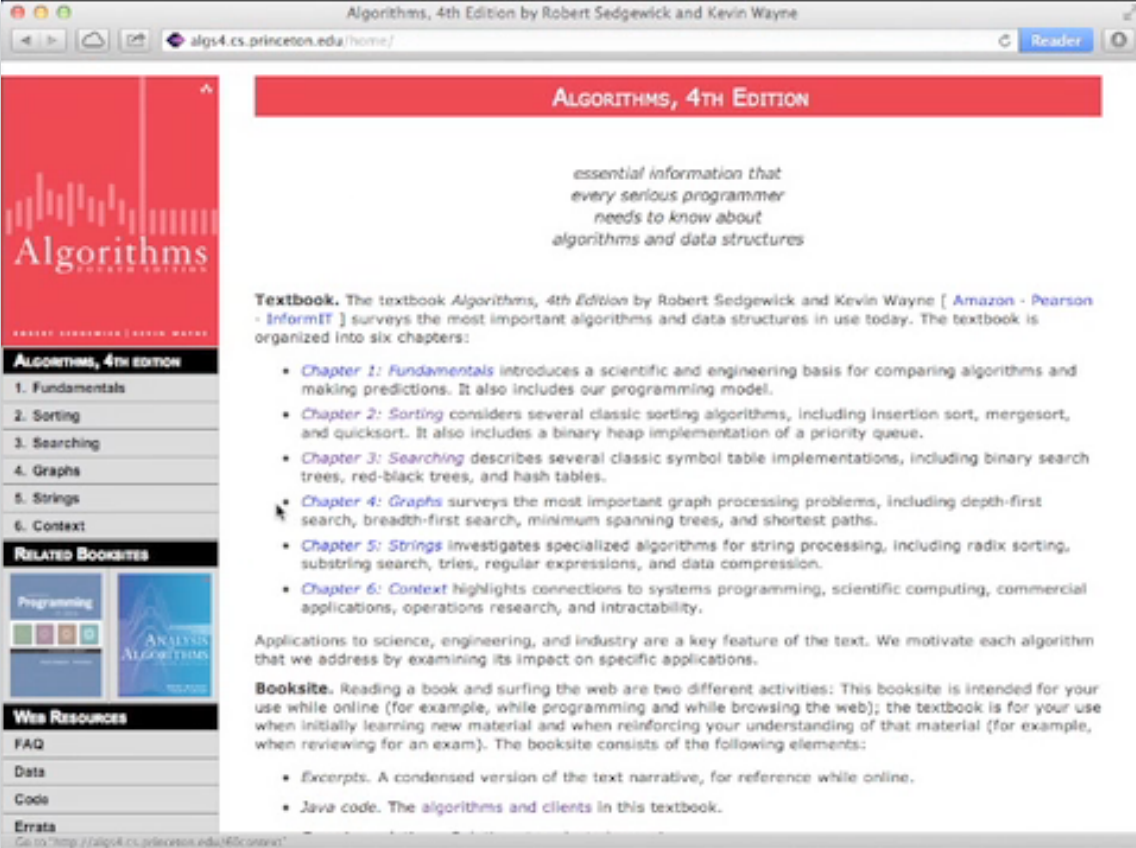
2010-2011 (4th iteration)

- *Lecture presentations.*
- *Assignments*
- *Exercises and Exams*
- *Precepts.*

Example: Algorithms booksite

Web contact associated with a book

- Web presence.
- Landing and takeoff for search.
- Code, test data, animations.
- Course materials.
- A living document.
- For use while computing, exploring.



The screenshot shows a web browser window displaying the homepage for the book "Algorithms, 4th Edition" by Robert Sedgwick and Kevin Wayne. The browser's address bar shows the URL "algs4.cs.princeton.edu/home/". The page features a red header with the title "ALGORITHMS, 4TH EDITION" and a quote: "essential information that every serious programmer needs to know about algorithms and data structures". Below the header, there is a section for the textbook, a list of chapters, a "RELATED BOOKSITES" section, and a "Web Resources" section. The "Web Resources" section includes links for "FAQ", "Data", "Code", and "Errata".

Algorithms, 4th Edition by Robert Sedgwick and Kevin Wayne

algs4.cs.princeton.edu/home/

ALGORITHMS, 4TH EDITION

essential information that every serious programmer needs to know about algorithms and data structures

Textbook. The textbook *Algorithms, 4th Edition* by Robert Sedgwick and Kevin Wayne [[Amazon](#) - [Pearson](#) - [InformIT](#)] surveys the most important algorithms and data structures in use today. The textbook is organized into six chapters:

- **Chapter 1: Fundamentals** introduces a scientific and engineering basis for comparing algorithms and making predictions. It also includes our programming model.
- **Chapter 2: Sorting** considers several classic sorting algorithms, including insertion sort, mergesort, and quicksort. It also includes a binary heap implementation of a priority queue.
- **Chapter 3: Searching** describes several classic symbol table implementations, including binary search trees, red-black trees, and hash tables.
- **Chapter 4: Graphs** surveys the most important graph processing problems, including depth-first search, breadth-first search, minimum spanning trees, and shortest paths.
- **Chapter 5: Strings** investigates specialized algorithms for string processing, including radix sorting, substring search, tries, regular expressions, and data compression.
- **Chapter 6: Context** highlights connections to systems programming, scientific computing, commercial applications, operations research, and intractability.

Applications to science, engineering, and industry are a key feature of the text. We motivate each algorithm that we address by examining its impact on specific applications.

Booksite. Reading a book and surfing the web are two different activities: This booksite is intended for your use while online (for example, while programming and while browsing the web); the textbook is for your use when initially learning new material and when reinforcing your understanding of that material (for example, when reviewing for an exam). The booksite consists of the following elements:

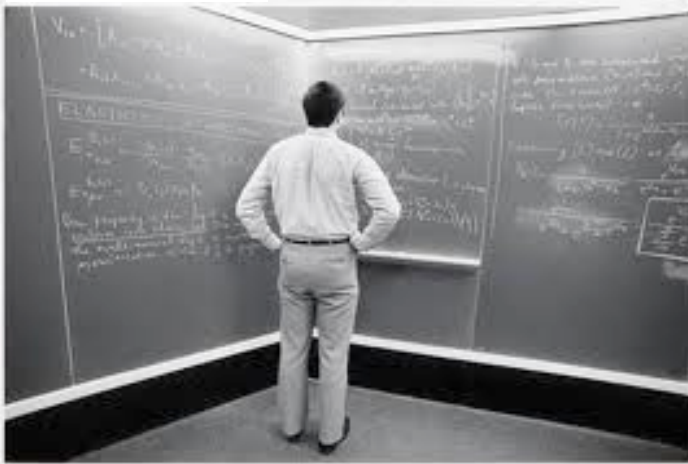
- **Excerpts.** A condensed version of the text narrative, for reference while online.
- **Java code.** The algorithms and clients in this textbook.

Go to: <http://algs4.cs.princeton.edu/40c/insert/>

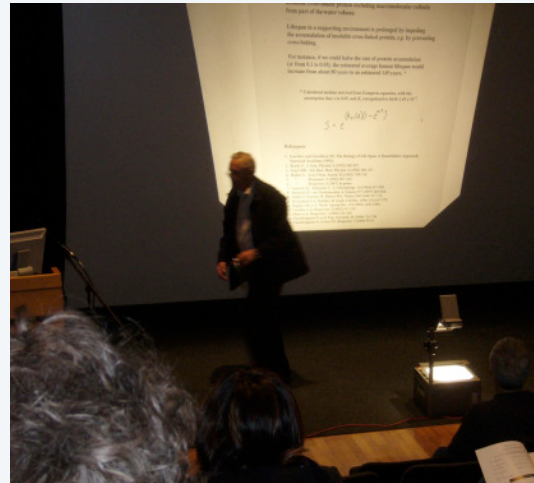
Lecture presentation materials

are evolving to new standard of excellence

Chalktalk



Overhead projection



Top-down 2-3-4 trees

- Transform tree on the way down to ensure that last node is not a 4-node

SINGLE ROTATION

$c^r.r := gc^r.l;$
 $y^r.l := gc;$
 $gc^r.r := c;$

"Split" any 4-nodes or one of two trans

Invariant: "current" node is not a 4-node
 Insertion at bottom therefore easy.

"Powerpoints"



This is a horse.

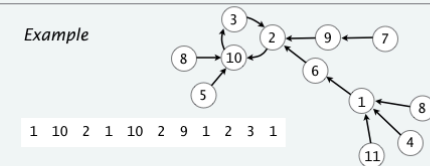
State of the art presentations (stay tuned)

Connected components in mappings

Class Y , the class of cycles of Cayley trees

EGF $Y(z) = \sum_{y \in Y} \frac{z^{|y|}}{|y|!} \equiv \sum_{N \geq 0} Y_N \frac{z^N}{N!}$

Example



Construction

$$Y = \text{CYC}(C)$$

"a component is a cycle of trees"

EGF equation

$$Y(z) = \ln \frac{1}{1 - C(z)}$$

Extract coefficients by Lagrange inversion with $f(u) = u/e^u$ and $H(u) = \ln(1/(1-u))$

$$[z^N]Y(z) = \frac{1}{N} [u^{N-1}] \frac{1}{1-u} e^{uN}$$

$$= \sum_{0 < k < N} \frac{N^{k-1}}{k!} = \sum_{1 < k < N} \frac{N^{N-k-1}}{(N-k)!}$$

$$Y_N = N! [z^N]Y(z) = N^{N-1} \sum_{1 < k < N} \frac{N!}{N^k (N-k)!} = N^{N-1} Q(N) \sim \frac{N^N \sqrt{\pi}}{\sqrt{2N}}$$

Lagrange Inversion Theorem (Bürmann form).

If a GF $g(z) = \sum_{n \geq 1} g_n z^n$ satisfies the equation $z = f(g(z))$ with $f(0) = 0$ and $f'(0) \neq 0$ then, for any function $H(u)$, $[z^N]H(g(z)) = \frac{1}{N} [u^{N-1}] H(u) \left(\frac{u}{f(u)} \right)^N$

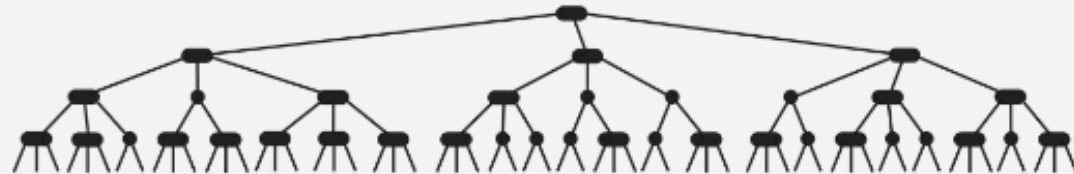
Example: An algorithms lecture

Elements

- Diagrams of data structures.
- Code.
- Animations.
- Summary Info.
- "Story".

2-3 tree: performance

Perfect balance. Every path from root to null link has same length.



Tree height.

- Worst case: $\lg N$. [all 2-nodes]
- Best case: $\log_3 N \approx .631 \lg N$. [all 3-nodes]
- Between 12 and 20 for a million nodes.
- Between 18 and 30 for a billion nodes.

Guaranteed **logarithmic** performance for search and insert.

Course materials

are dramatically improved via web dissemination

Detailed assignments

Online discussion

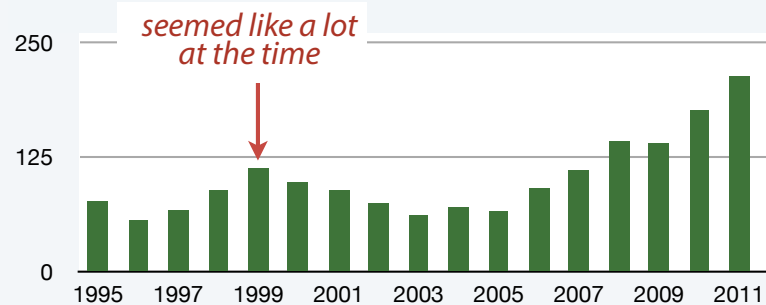
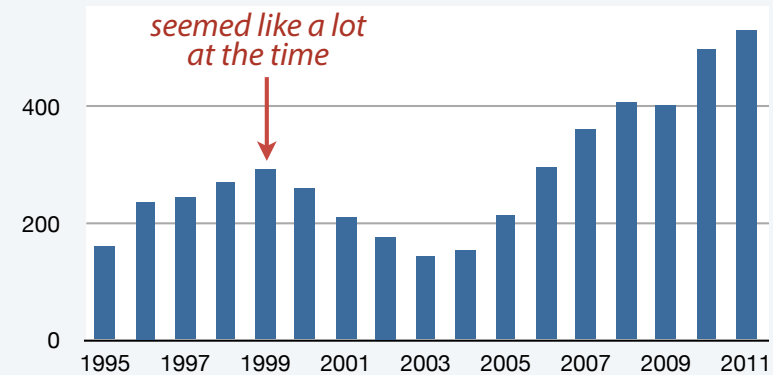
Electronic submission and assessment

Schedule of events

2011: Time to declare victory?

Introduction to CS enrollments

- *Double* the height of the “bubble”
- 60% of all Princeton students.



“Algorithms” enrollments

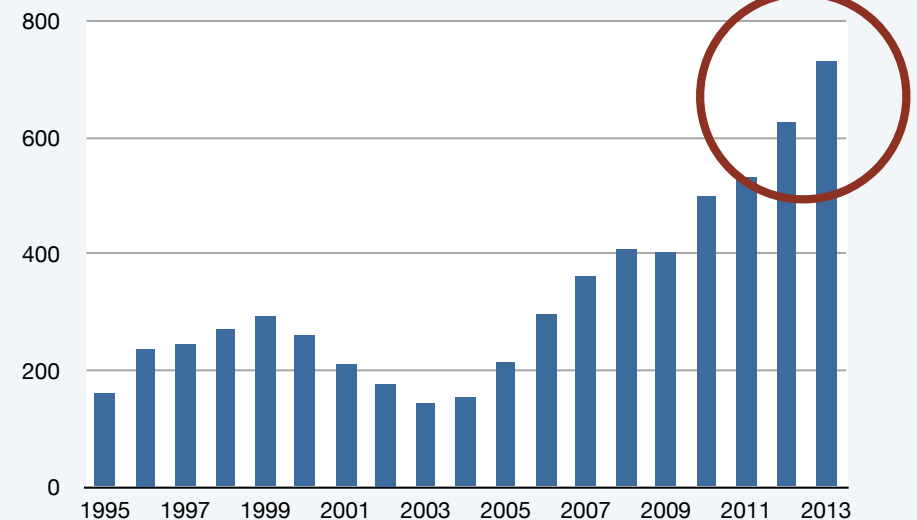
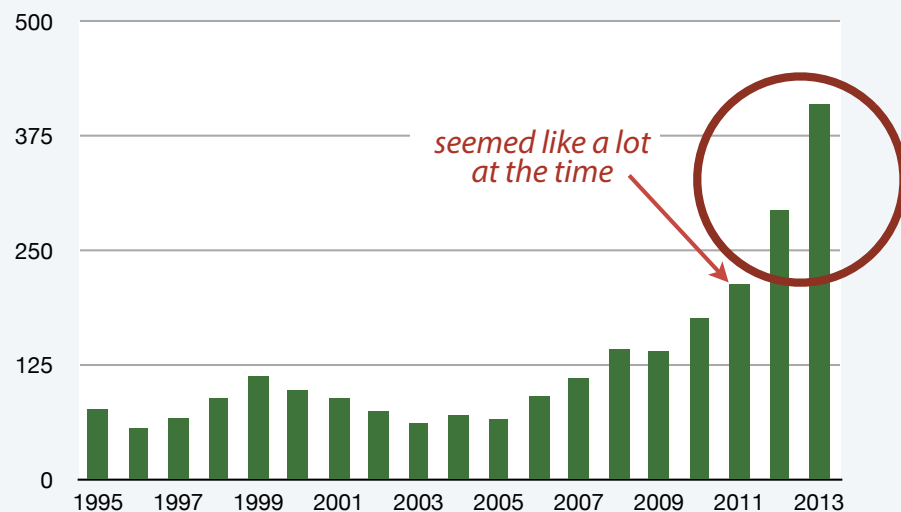
- *Three times* the height of the “bubble”
- 35% of all Princeton students.

Q. (2011) When will enrollments start to decline?

2013: Time to declare victory? Not yet!

Introduction to CS enrollments

- *Triple* the height of the “bubble”
- 60% of all Princeton students.



“Algorithms” enrollments

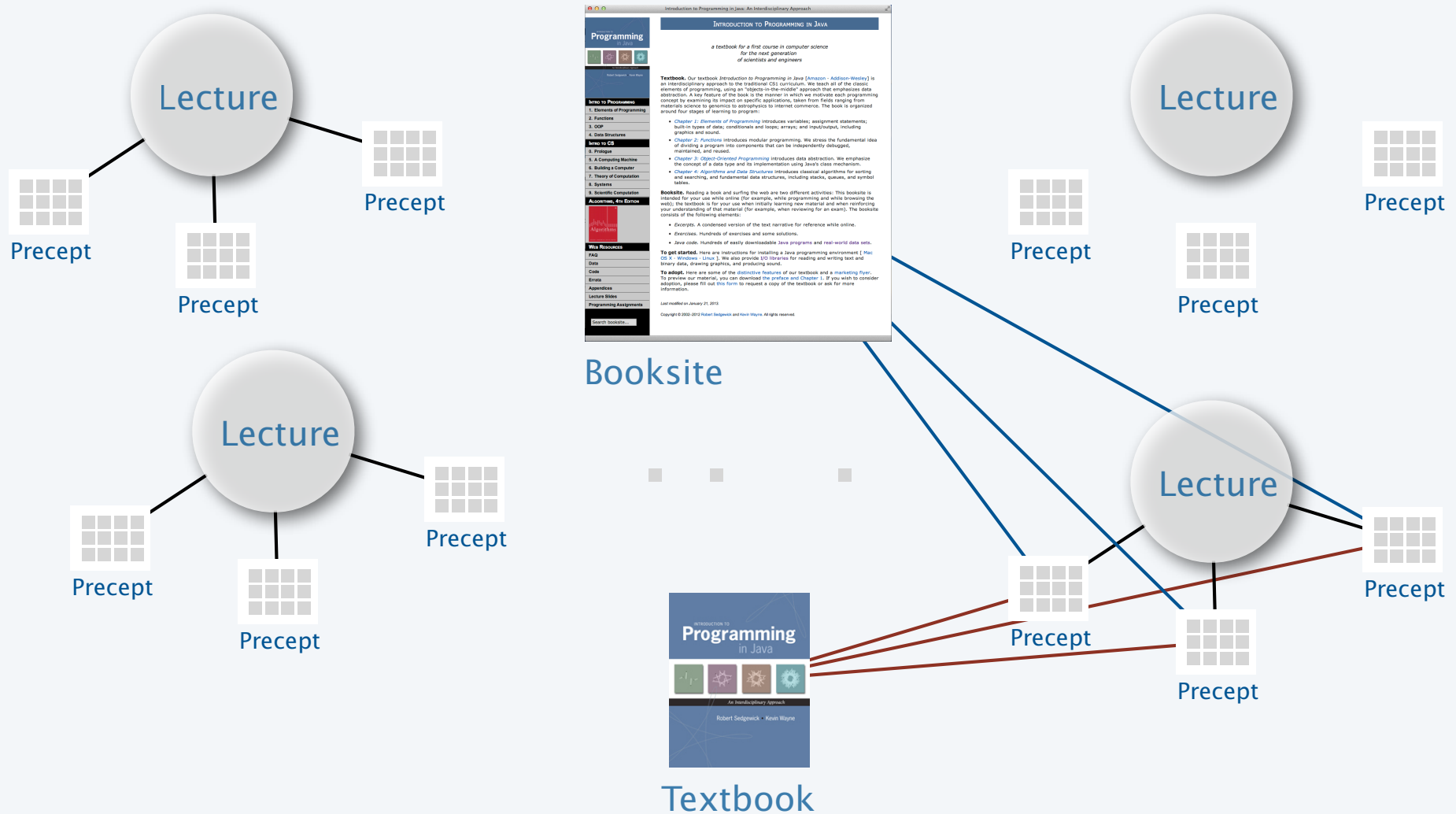
- *Four times* the height of the “bubble”.
- *Doubled* in the last two years.
- 35% of all Princeton students.

Q. (2011) When will enrollments start to decline?

A. (2013) Sometime after they stop *accelerating*!

Primary advantage of booksite model

is *scalability* allowing reach to at least an order of magnitude more students



Handles huge numbers of students worldwide.
Instructors use book and booksite as basis for teaching.
Individuals are directly accessing book and booksite for self-study.

Next challenge (2011)

RS. *Algorithms for the masses* (ANALCO, San Francisco 2011)

Summary

The scientific method is an essential ingredient in programming.
Embracing, supporting, and leveraging science in
intro CS and algorithms courses can serve large numbers of students.

Proof of concept: First-year courses at Princeton

- 40+% of Princeton students in a single intro course
- 25+% of Princeton students in a single **algorithms** course

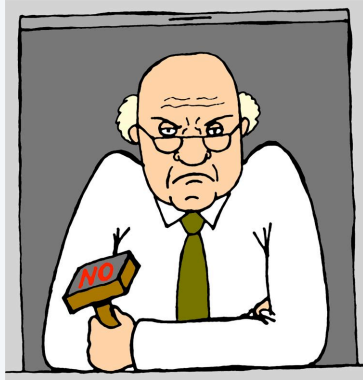
Next goals:

- 60+% of **all college students** in an intro CS course
- 35+% of **all college students** in an **algorithms** course

↑
Algorithms for the masses

Confession: No idea how we would get there...

2012: MOOCs go mainstream



*With apologies to
our actual
administrators*

Q. Are you interested in teaching online?

RS. No. (Too much work to do it properly.)

Q. Trustees want it, so we're doing it anyway. Are you in?

RS+KW. An offer we cannot refuse...



Andrew Ng and Daphne Koller

The logo for Coursera, featuring the word "coursera" in a blue, lowercase, sans-serif font. The letter "o" is stylized with a circular arrow around it.

An online platform for the "course" abstraction

Immediate realization:
Our model is *perfectly suited* to go online.

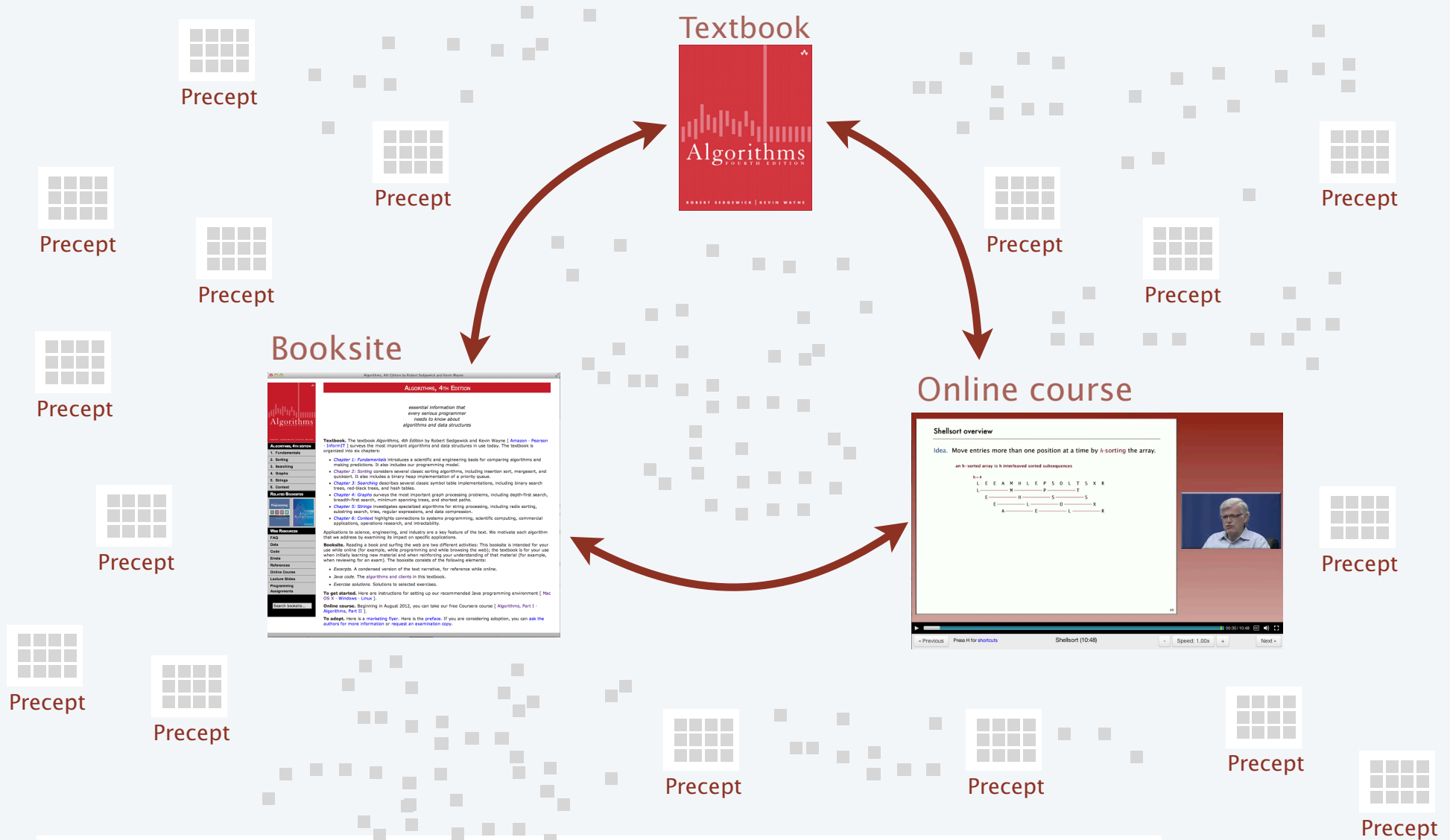
Details are another story...

"Algorithms, Part I" (Fall 2012)

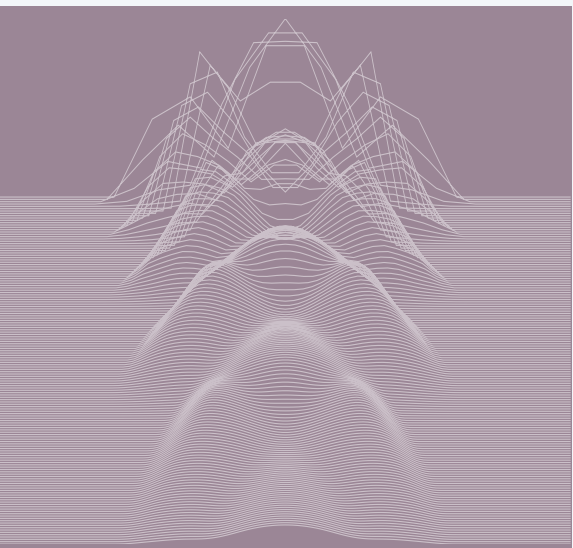
A screenshot of a video lecture. The main content is a slide titled "Shellsort overview". The slide text includes: "Idea. Move entries more than one position at a time by h-sorting the array.", "an h-sorted array is h interleaved sorted subsequences", and a diagram showing the interleaving of subsequences for h=4. The diagram shows the sequence "L E E A M H L E P S O L T S X R" with lines connecting letters to their positions in the sorted subsequences. A small video inset in the bottom right corner shows a man with glasses speaking. At the bottom of the screenshot is a video player interface with a progress bar, a "Speed: 1.00x" control, and navigation buttons for "Previous" and "Next".

Online teaching

extends our reach to at least *another* order of magnitude more students



Handles huge numbers of students worldwide.
Instructors use book, *course*, and booksite as basis for teaching.
Individuals are directly accessing all three for self-study.



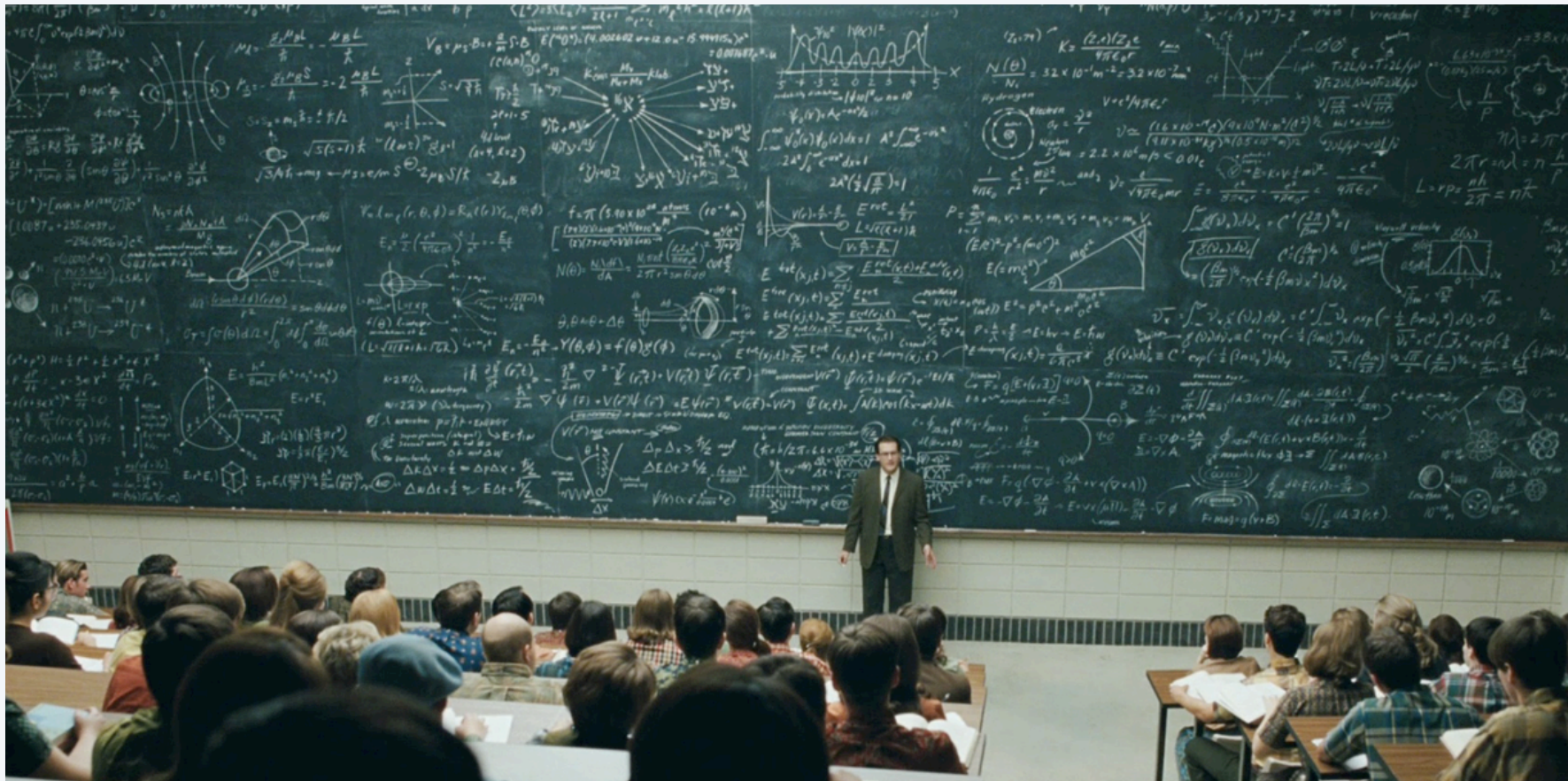
Analytic Combinatorics for the masses

- Frequently asked questions
- Disruptive changes
- A way forward
- **Analytic combinatorics**
- Questions answered

Analytic combinatorics

RS: Hey, maybe this could work for "Analytic Combinatorics" !

Prevailing wisdom: *No way—interactive chalktalk* is time-honored and needed for math.



RS: Seriously?

Presentation elements: Analytic Combinatorics

Mathematical derivations

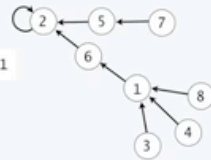
Cayley trees

Class \mathcal{C} , the class of labeled rooted unordered trees

EGF
$$C(z) = \sum_{c \in \mathcal{C}} \frac{z^{|c|}}{|c|!} \equiv \sum_{N \geq 0} C_N \frac{z^N}{N!}$$

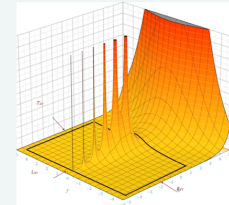
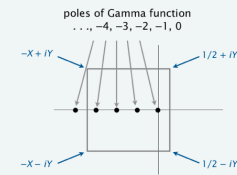
Example

6 2 1 1 2 2 5 1



52

Plots



"Story"

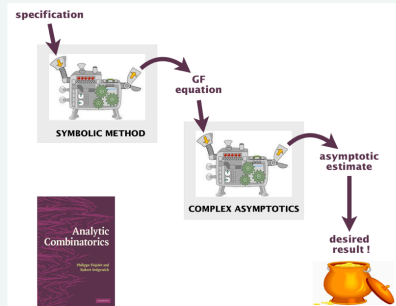
Analytic combinatorics overview

A. SYMBOLIC METHOD

1. OGFs
2. EGFs
3. MGFs

B. COMPLEX ASYMPTOTICS

4. Rational & Meromorphic
5. Applications of R&M
6. Singularity Analysis
7. Applications of SA
8. Saddle point

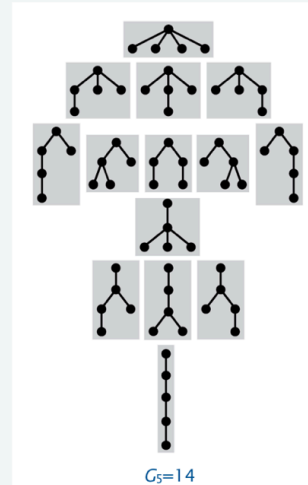
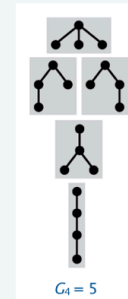
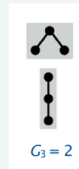


5

Drawings of combinatorial objects

Classic example of the symbolic method

Q. How many trees with N nodes?



Example: An Analytic Combinatorics Presentation

Analytic combinatorics: How many trees with N nodes?

Symbolic method

Combinatorial class

G , the class of all trees

Construction

$$G = \bullet \times \text{SEQ}(G)$$

"a tree is a node and a sequence of trees"

OGF equation

$$G(z) = z(1 + G(z) + G(z)^2 + G(z)^3 + \dots) = \frac{z}{1 - G(z)}$$

$$G(z) - G(z)^2 = z$$

Quadratic equation

$$G(z) = \frac{1 + \sqrt{1 - 4z}}{2}$$

Classic next steps

Binomial theorem

$$G(z) = -\frac{1}{2} \sum_{N \geq 1} \binom{\frac{1}{2}}{N} (-4z)^N$$

Extract coefficients

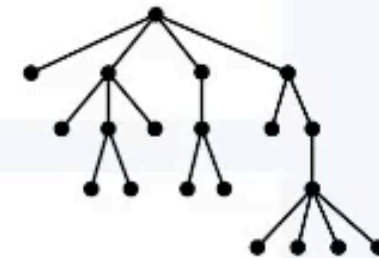
$$G_N = -\frac{1}{2} \binom{\frac{1}{2}}{N} (-4)^N = \frac{1}{N} \binom{2N-2}{N-1} = \frac{1}{4N-2} \binom{2N}{N}$$

Stirling's approximation

$$\sim \frac{1}{4N} \exp(2N \ln(2N) - 2N + \ln \sqrt{4\pi N} - 2(N \ln(N) - N + \ln \sqrt{2\pi N}))$$

Simplify

$$G_N \sim \frac{4^{N-1}}{\sqrt{\pi N^3}}$$



detailed calculations omitted

"An Introduction to Analytic Combinatorics" Lectures

Part I: Analysis of Algorithms

1. Introduction
2. Recurrences
3. Generating Functions
4. Asymptotic Analysis
5. Analytic combinatorics
6. Trees
7. Permutations
8. Strings and Tries
9. Words and Mappings

~500 slides

Part II: Analytic Combinatorics (in development)

1. Ordinary GFs
2. Exponential GFs
3. Bivariate GFs
4. Meromorphic Asymptotics
5. MA applications
6. Singularity Analysis
7. SA Applications
8. Saddle Point
9. Advanced Topics

~500 slides

2013: Mission accomplished

web content

aofa.cs.princeton.edu

text digests

code

exercise solutions

lecture slides

ac.cs.princeton.edu

to launch spring 2013

Introduction to the Analysis of Algorithms by Robert Sedgewick and Philippe Flajolet

aofa.cs.princeton.edu/home/

AN INTRODUCTION TO THE ANALYSIS OF ALGORITHMS

People who analyze algorithms have double happiness. First of all they experience the sheer beauty of elegant mathematical patterns that surround elegant computational procedures. Then they receive a practical payoff when their theories make it possible to get other jobs done more quickly and more economically. **D. E. Knuth**

This booksite is under development (Spring 2012). No promises.

Textbook. The textbook *An Introduction to the Analysis of Algorithms* by Robert Sedgewick and Philippe Flajolet [[Amazon](#) · [Inform IT](#)] overviews the primary techniques used in the mathematical analysis of algorithms. The material covered draws from classical mathematical topics, including discrete mathematics, elementary real analysis, and combinatorics, as well as from classical computer science topics, including algorithms and data structures.

- **Chapter 1: Analysis of Algorithms** considers the general motivations for algorithmic analysis and relationships among various approaches to studying performance characteristics of algorithms.
- **Chapter 2: Recurrence Relations** concentrates on fundamental mathematical properties of various types of *recurrence relations* which arise frequently when analyzing an algorithm through a direct mapping from a recursive representation of a program to a recursive representation of a function describing its properties.
- **Chapter 3: Generating Functions** introduces a central concept in the average-case analysis of algorithms: *generating functions* — a necessary and natural link between the algorithms that are our objects of study and analytic methods that are necessary to discover their properties.
- **Chapter 4: Asymptotic Approximations** examines methods of deriving approximate solutions to problems or of approximating exact solutions, which allow us to develop *concise and precise* estimates of quantities of interest when analyzing algorithms.
- **Chapter 5: Trees** investigates properties of many different types of *trees*, fundamental structures that arise implicitly and explicitly in many practical algorithms. Our goal is to provide access to results from an extensive literature on the combinatorial analysis of trees, while at the same time providing the groundwork for a host of algorithmic applications.
- **Chapter 6: Permutations** surveys combinatorial properties of permutations (orderings of the numbers 1 through n) and shows how they relate in a natural way to fundamental and widely-used sorting algorithms.
- **Chapter 7: String and Tries** studies basic combinatorial properties of *strings*, sequences of characters or letters drawn from a fixed alphabet, and introduces algorithms that process strings ranging from fundamental methods at the heart of the theory of computation to practical text-processing methods with a host of important applications.
- **Chapter 8: Words and Maps** covers global properties of *words* (N -letter strings from an N -letter alphabet), which are well-studied in classical combinatorics (because they model sequences of independent Bernoulli trials) and in classical applied algorithmics (because they model input sequences for hashing algorithms). The chapter also covers random *maps* (N -letter words from an N -letter alphabet) and discusses relationships with trees and permutations.

Booksite. Reading a book and surfing the web are two different activities: This booksite is intended for your use while online (for example, while programming and while browsing the web); the textbook is for your use when initially learning new material and when reinforcing your understanding of that material (for example, when reviewing for an exam). The booksite consists of the following elements:

- **Excerpts.** A condensed version of the text narrative, for reference while online.

online course

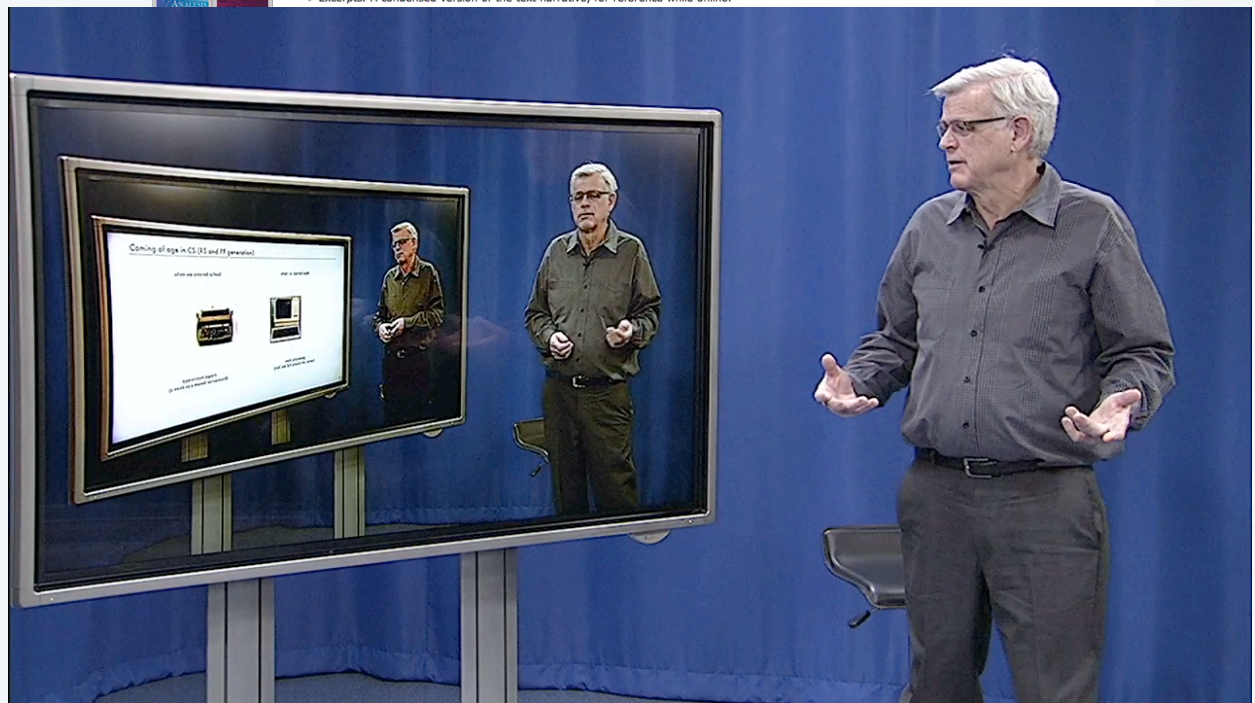
*Analytic
Combinatorics*

February-April 2013

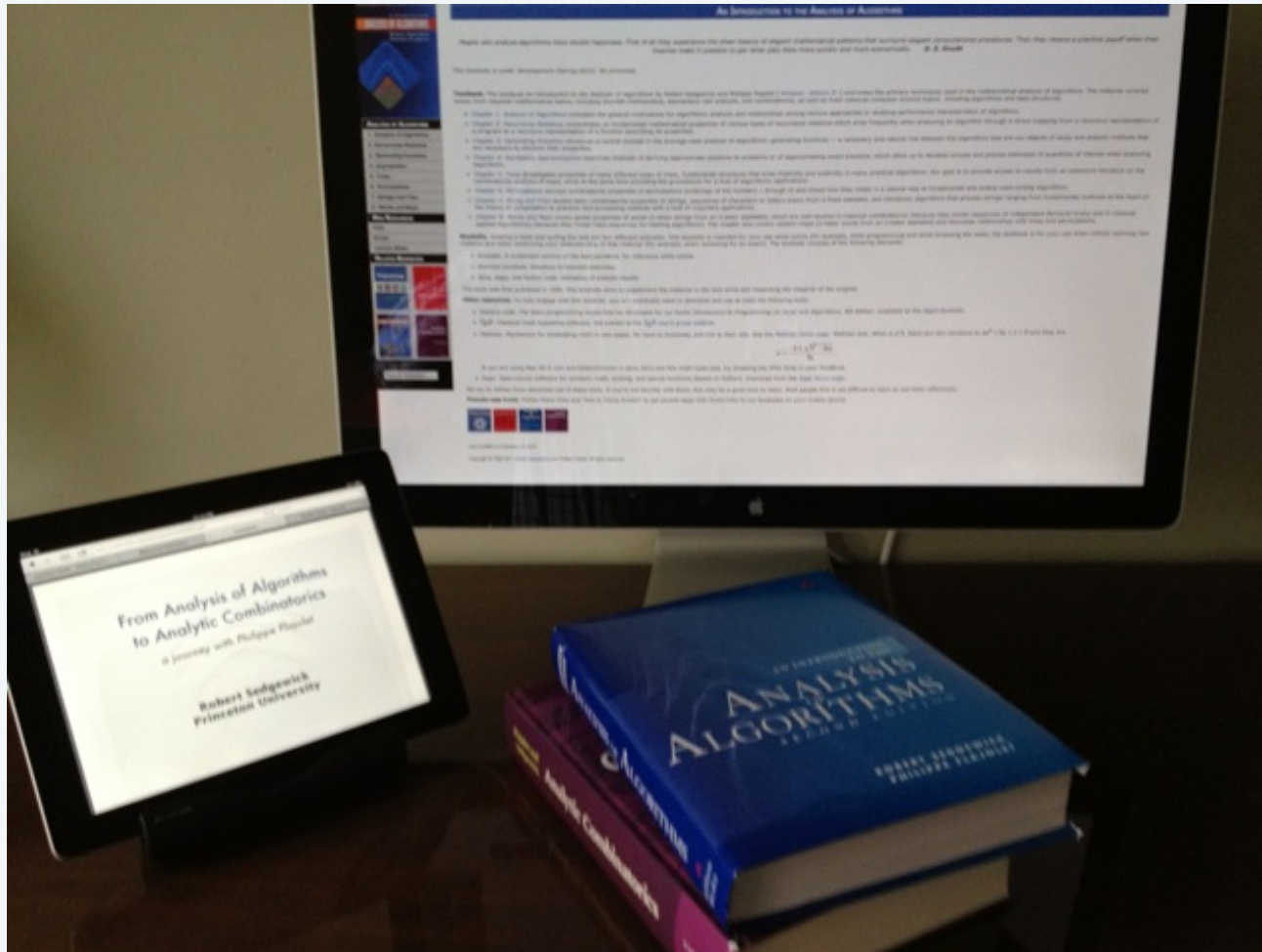
10 lectures on AofA

10 lectures on AC

25,000+ registrants



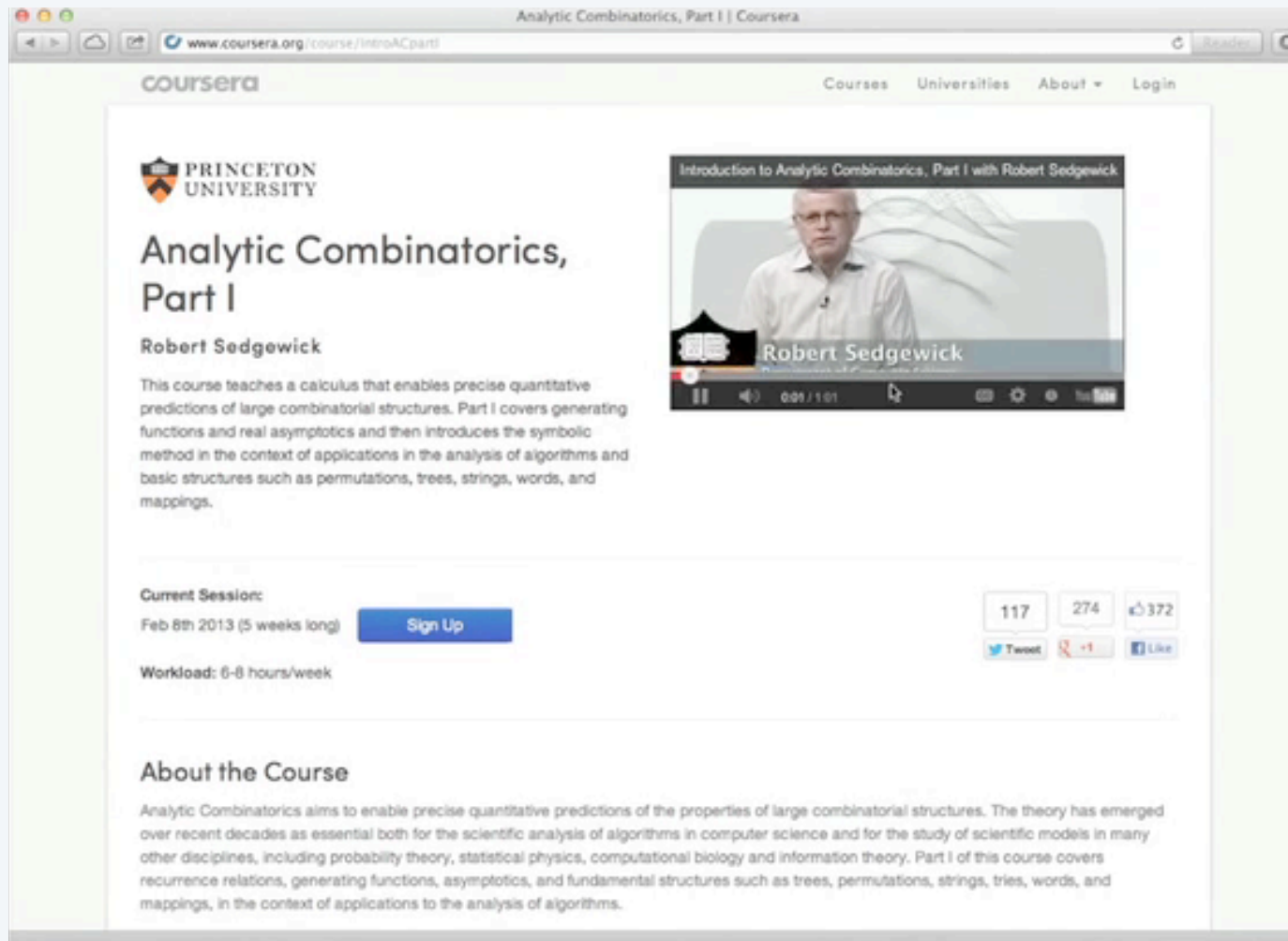
2013: Mission accomplished!



Q. Can an advanced subject such as *Analytic Combinatorics* be taught effectively online?

A. Absolutely! Indeed, advanced subjects may be a *sweet spot* for MOOCs.

How do I take an online course?



The screenshot shows a web browser window with the URL www.coursera.org/course/introACpart. The page header includes the Coursera logo and navigation links for Courses, Universities, About, and Login. The main content area features the Princeton University logo and the course title "Analytic Combinatorics, Part I" by Robert Sedgewick. A video player is embedded, showing a video titled "Introduction to Analytic Combinatorics, Part I with Robert Sedgewick" with a progress bar at 0:01 / 1:01. Below the video, the course description states: "This course teaches a calculus that enables precise quantitative predictions of large combinatorial structures. Part I covers generating functions and real asymptotics and then introduces the symbolic method in the context of applications in the analysis of algorithms and basic structures such as permutations, trees, strings, words, and mappings." The "Current Session" is listed as "Feb 8th 2013 (5 weeks long)" with a "Sign Up" button. The workload is "6-8 hours/week". Social media sharing options for Twitter (117), +1 (274), and Like (372) are displayed. The "About the Course" section provides further details on the course's focus on combinatorial structures and their applications in computer science and other disciplines.

coursera Courses Universities About Login

PRINCETON UNIVERSITY

Analytic Combinatorics, Part I

Robert Sedgewick

This course teaches a calculus that enables precise quantitative predictions of large combinatorial structures. Part I covers generating functions and real asymptotics and then introduces the symbolic method in the context of applications in the analysis of algorithms and basic structures such as permutations, trees, strings, words, and mappings.

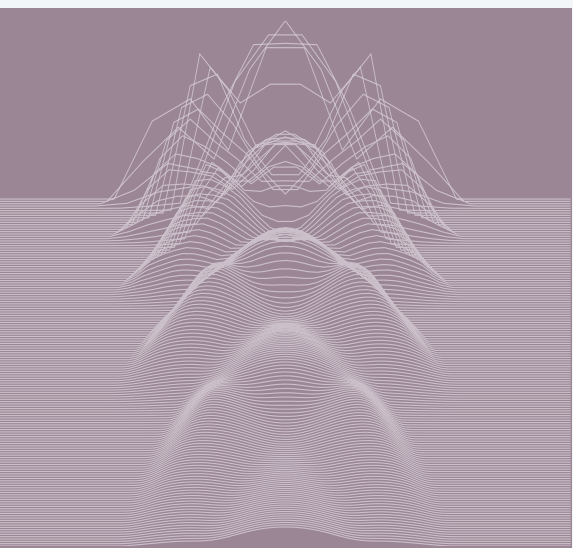
Current Session:
Feb 8th 2013 (5 weeks long) [Sign Up](#)

Workload: 6-8 hours/week

About the Course

Analytic Combinatorics aims to enable precise quantitative predictions of the properties of large combinatorial structures. The theory has emerged over recent decades as essential both for the scientific analysis of algorithms in computer science and for the study of scientific models in many other disciplines, including probability theory, statistical physics, computational biology and information theory. Part I of this course covers recurrence relations, generating functions, asymptotics, and fundamental structures such as trees, permutations, strings, tries, words, and mappings, in the context of applications to the analysis of algorithms.

117 274 372
Tweet +1 Like



Analytic Combinatorics for the masses

- Frequently asked questions
- Disruptive changes
- A way forward
- Analytic combinatorics
- **Questions answered**

Questions answered

Q. What is a MOOC?

A. An *irresistible opportunity* for individuals and institutions who want to disseminate knowledge to *vastly expand* their reach.

Q. Can an advanced subject such as *Analytic Combinatorics* be taught effectively online?

A. Absolutely! Indeed, such subjects may be a *sweet spot* for MOOCs.

Q. Why?

A. *Anyone* interested in disseminating knowledge can put a good course online nowadays.

A. No university can afford to teach *all* advanced subjects.

Questions answered



*With apologies to
our actual students*

Q. How much does it cost for me to take a MOOC?

A. It *must* be *free*. If not, it's not a MOOC.

Q. Can I get credit for taking a MOOC?

A. For *free*? Not likely.

A. You can pay to take a test.

A. You can take a MOOC as part (or all) of a course at a university.

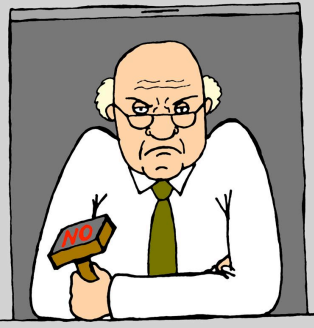
Observation: The vast majority of MOOC students don't care!

Q. If you're giving away content, what am I getting for the money at University X?

A. We make sure that you learn as much as you are able.

A. We certify that you did so.

Questions answered



Q. Can we save money by moving our teaching online?

A. NO.

A. You can *improve the quality of your teaching and extend your reach.*

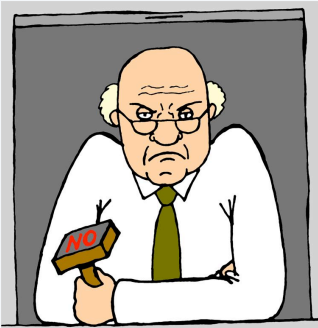
Q. How?

A. Make teachers first-class citizens.

A. Remove disincentives for researchers who choose to teach.

A. Provide support and real incentives for teaching and content creation.

Questions answered



Q. How will this impact our current students?

A. Many online courses will be better than the ones you now offer.

A. Focus will evolve to communities of scholars led by teachers working with presentations possibly developed somewhere else.

A. Everyone's education will have a significant online component.

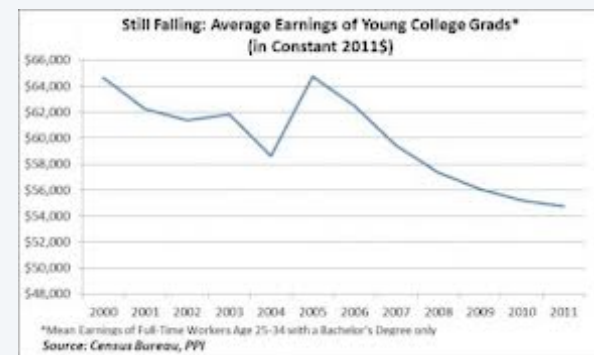
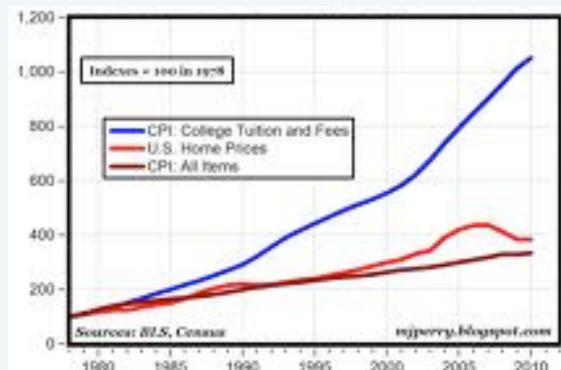
Q. How much will it cost our institution to embrace online education?

A. Less than you are spending on many things that are less central to your mission.

A. You need to plan to invest in online at the scale you are investing in the library.

A. Can you afford to *not* embrace online education?

**CPI: College Tuition vs.
U.S. Home Prices vs.
CPI: All Items, 1978 to 2010**



A parting thought

(from *John Hennessy* in an interview for an article by Ken Auletta the New Yorker, 2012)



“[Universities,] like newspapers and music companies and much of traditional media a little more than a decade ago are sailing in seemingly placid waters.”

“But ... *there's a tsunami coming.*”





Analytic Combinatorics for the masses

Robert Sedgewick
Princeton University