

A 21st Century Model for Disseminating Knowledge

Robert Sedgewick
Princeton University

[joint work with Kevin Wayne]

Underlying themes

CS is in crisis! What's the plan?

- CS faculty carrying 2-3 times the load of everyone else.
- Best and brightest have great opportunities in industry.
- National Academies report recommends that institutions *respond with urgency* but all are too exhausted to do it.



CS for all (really).

- Many, many students are learning how to code.
- Relatively few are learning *computer science*.

You're using my textbook; why not use my lectures?

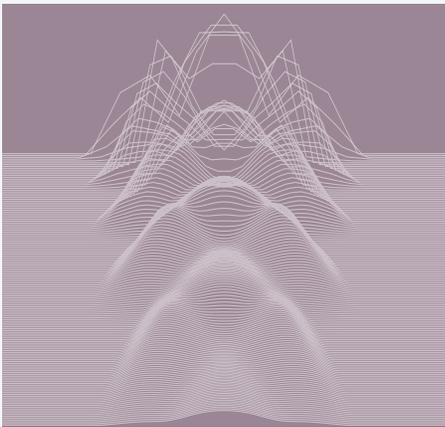
- Many faculty are preparing and giving the *same* lectures.
- Live large lecture format known to be flawed; why use it?
- Well-produced online lectures can address both problems.



Netflix for education: on the horizon?

- Access to education when and where you want it is an extremely powerful idea.
- Millions of content creators (professors); billions of consumers (learners).
- Missing link: 21st century marketplace

A 21st Century Model for Disseminating Knowledge

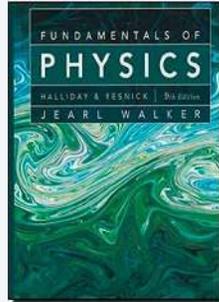


- **Mission accomplished?**
- Disruptive changes
- Taking the plunge
- A way forward
- Postscript

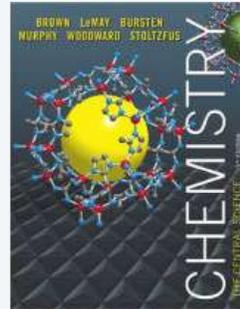
Fundamental challenge for teaching CS (1965-present)

has been to do without a *standard textbook* (the norm in other fields).

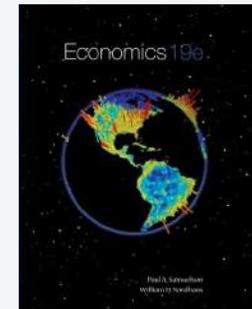
Physics



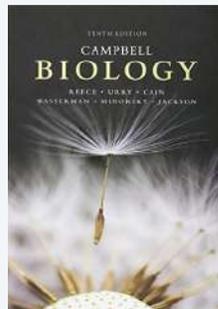
Chemistry



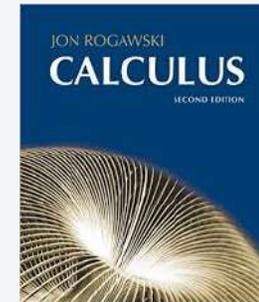
Economics



Biology



Calculus



**Computer
Science**



Central thesis (RS, 1975)

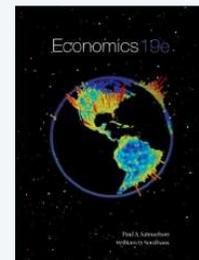
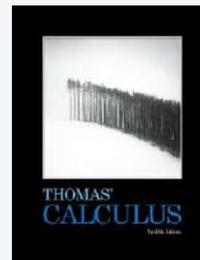
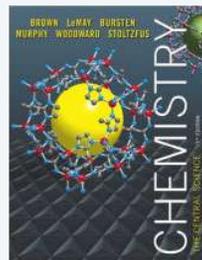
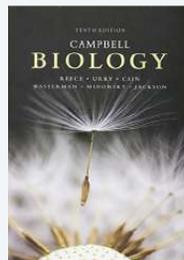
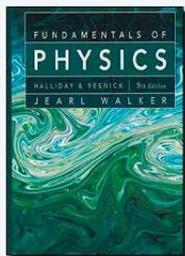
Most college students need a course in **algorithms**

The study of algorithms is a significant body of knowledge that is

- intellectually challenging
- pervasive in modern life
- critical to modern science and engineering

Anyone wanting to use a computer effectively needs to understand

- the scientific method in understanding program behavior
- how to compare algorithms and predict performance
- data abstraction
- classic data structures and types
- applying them to solve modern problems



Goal: A *textbook* for teaching algorithms to scientists, engineers and programmers.

Mission accomplished (2011)?

Algorithms, Fourth Edition

Classic text for decades, 750,000+ sold.

- “Algorithms with code”.
- Modern programming model.
- Model course in ACM–IEEE curriculum.
- Completely revamped each decade.
- Widely used around the world.
- Found throughout software infrastructure.



| | <i>edition</i> | <i>goal for code: clear, readable and</i> |
|-------|----------------|---|
| 1970s | 1st | <i>compiles</i> |
| 1980s | 2nd | <i>runs on examples</i> |
| 1990s | 3rd | <i>meets performance specs</i> |
| 2010s | 4th | <i>industrial strength</i> |

Problem: Many students are *not prepared* to learn much of the material.

Central thesis revised (RS, 1992)

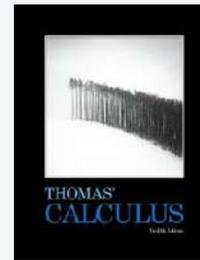
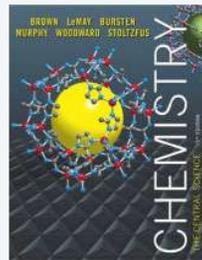
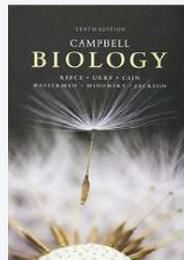
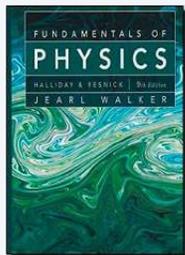
All college students need courses in **computer science**

Computer science embraces a significant body of knowledge that is

- intellectually challenging
- pervasive in modern life
- critical to further study in most (if not all) academic disciplines

Anyone can learn the importance of

- modern programming models through data abstraction
- the scientific method in understanding program behavior
- algorithms and data structures
- abstract machines and their connections to real ones
- computation in a broad variety of applications
- preparing for a lifetime of engaging with computation



Goal: A *standard intro text* for CS that can stand alongside other standard intro texts.

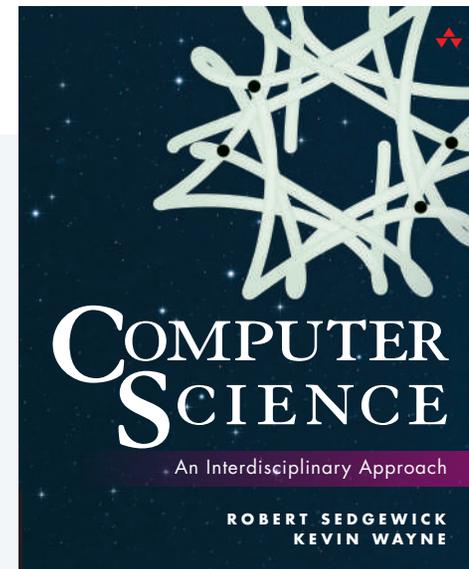
[decades of difficult challenges omitted.]

- Identify content
- Change content
- Interface with the computer center
- Choose programming language
- Dot-com bust (enrolls way down)
- Change programming language
- Staffing
- Political battles
- Competing courses
- Inadequate resources
- Abandon computer center
- Financial crash (enrolls way up)
- Windows, OS X, Linux
- ...

Mission accomplished? (2016)

Computer Science: An Interdisciplinary Approach

- 25 years in development.
- Fundamental ideas in the field for first-year students.
- Programming in Java through data abstraction.
- Theory of computation from Turing machines to intractability.
- Machine architecture and circuit design from gates to machine language.
- Turing, von Neumann, Boole, Shannon, . . .
- Stands with intro courses in economics, physics, biology, and other disciplines.
- Basis for Princeton's most popular course.
- Model course in ACM-IEEE curriculum.



Interdisciplinary approach that embraces, leverages and supports other disciplines.

All in the context of applications

Ideal programming example/assignment

- teaches a basic CS concept
- solves an important problem
- is intellectually engaging and appealing
- is open-ended

```
public class BouncingBall
{
    public static void main(String[] args)
    { // Simulate the movement of a bouncing ball.
        double rx = .480, ry = .860;
        double vx = .015, vy = .023;
        double radius = .05;

        StdDraw.setXscale(-1.0, +1.0);
        StdDraw.setYscale(-1.0, +1.0);

        while(true)
        { // Update ball position and draw it there.
            if (Math.abs(rx + vx) + radius > 1.0) vx = -vx;
            if (Math.abs(ry + vy) + radius > 1.0) vy = -vy;

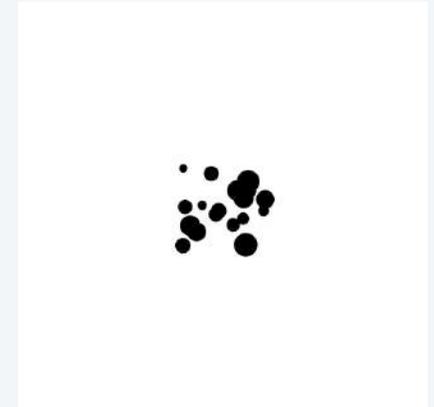
            rx = rx + vx;
            ry = ry + vy;

            StdDraw.clear();
            StdDraw.setPenColor(StdDraw.BLACK);
            StdDraw.filledCircle(rx, ry, radius);
            StdDraw.show(20);
        }
    }
}
```

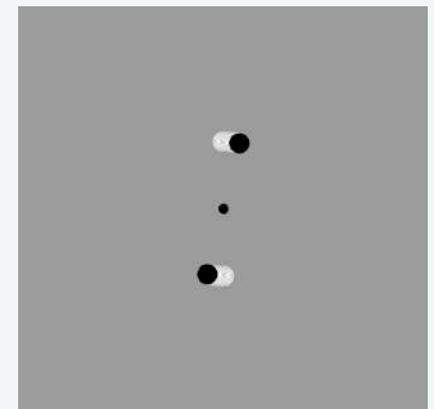
Bouncing ball



Bouncing balls

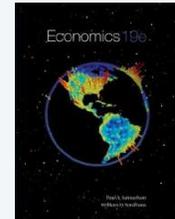
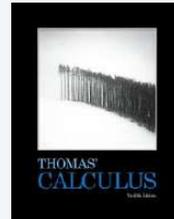
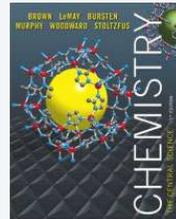
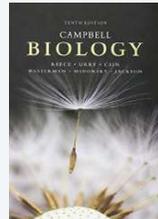
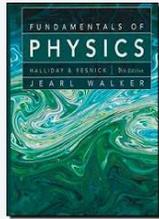


N-body



2011: Time to declare victory?

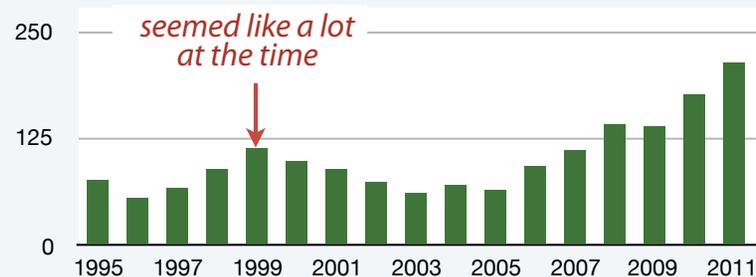
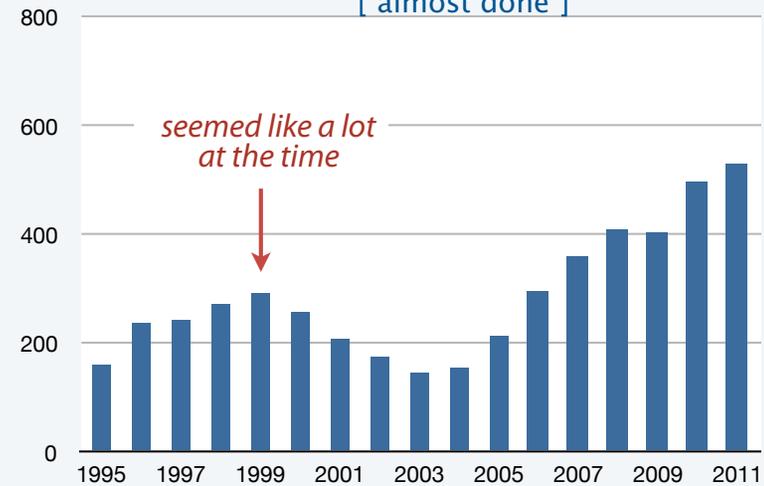
Goal: A *standard intro text* for CS that can stand alongside other standard intro texts.



["almost done"]

Introduction to CS enrollments

- *Double* the height of the “bubble”
- 43% of all Princeton students.



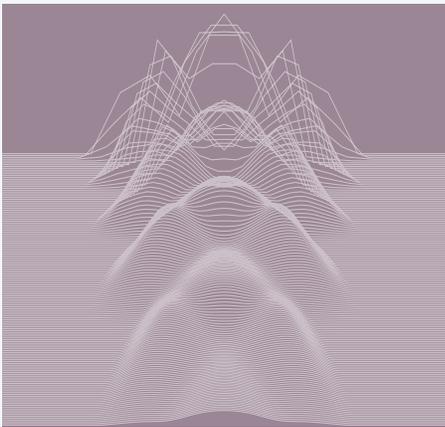
“Algorithms” enrollments

- *Three times* the height of the “bubble”
- 17% of all Princeton students.

Little did we know what the future held...



A 21st Century Model for Disseminating Knowledge



- Mission accomplished?
- **Disruptive changes**
- Taking the plunge
- A way forward
- Postscript

Disruptive change I: Libraries

FUTURE READING: Digitization and its discontents.
by Anthony Grafton

The New Yorker
November 5, 2007



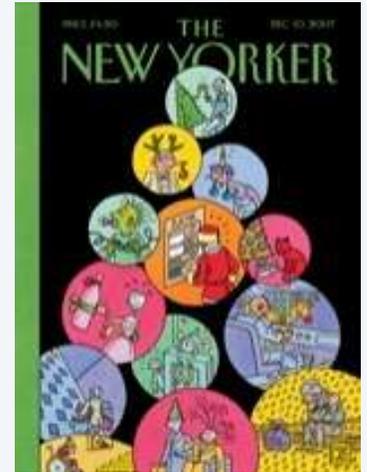
Sit in your local coffee shop, and your laptop can tell you a lot. If you want deeper, more local knowledge, you will have to take the narrower path that leads between the lions and up the stairs.

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

Letter to the editor
by Robert Sedgewick

The New Yorker
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.

Future of libraries?

20th century

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



Early 21st century

- 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?

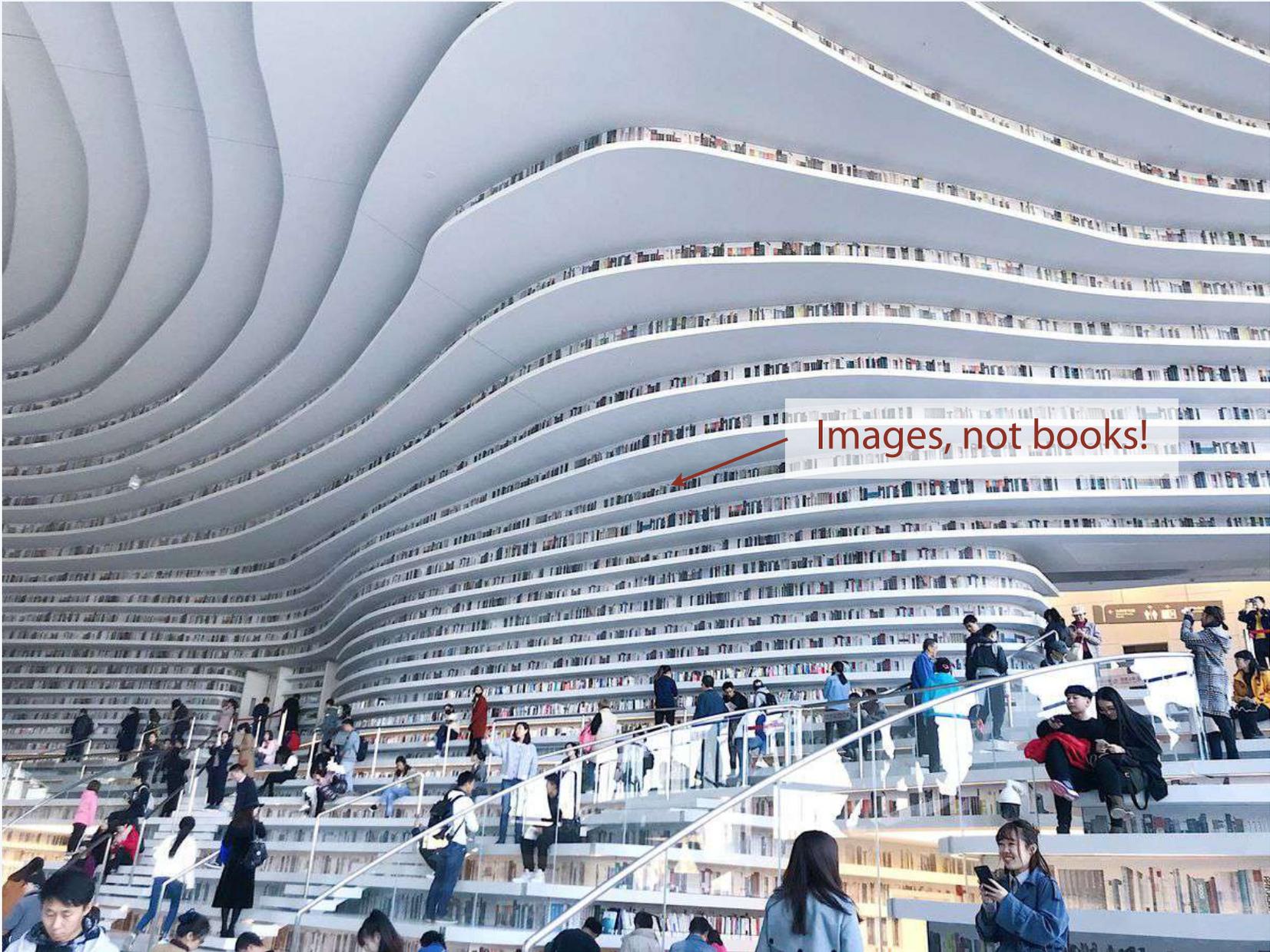
- A few book museums (for Grafton)
- Digital library infrastructure (for everyone else)
- Use old libraries as office space for CS faculty (!)



How will we disseminate knowledge in the future?
Will universities play a role?



Futuristic Binhai Library in Tianjin, China



Images, not books!

A book-lover's dream . . . NOT!

Disruptive change 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

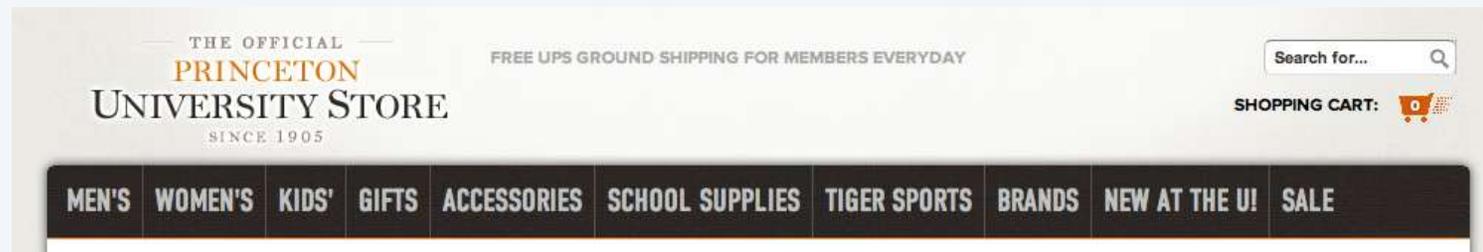
600,000
Decline in number of
new college textbooks
sold, 2004-5



Princeton
U-store
1950s



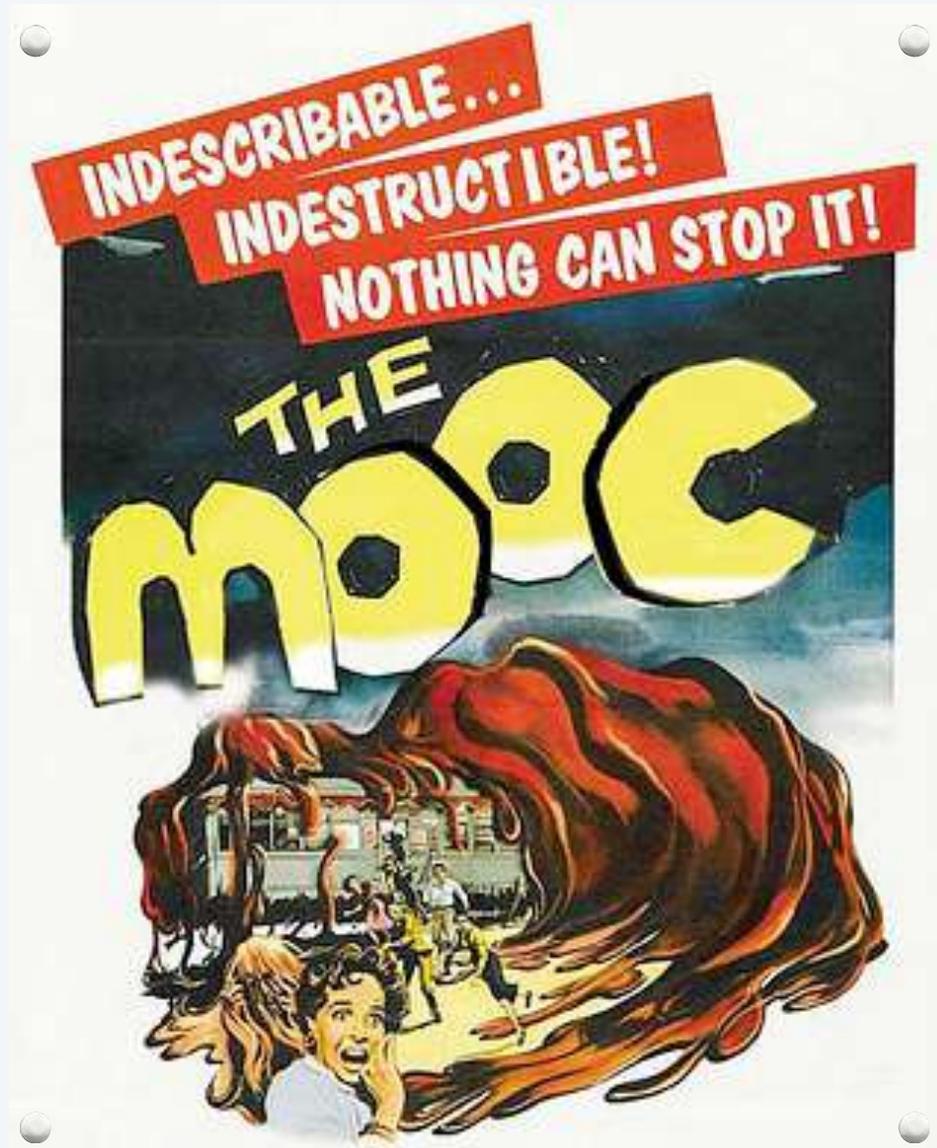
Princeton
U-store
2010s



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

↑
No books!

Disruptive change III



2012: MOOCs go mainstream



With apologies to
our actual
administrators

Q. (Jan. 2012) Are you interested in teaching online?
RS+KW. No. (Too much work to do it properly.)

Q. (Apr. 2012) Trustees want it: 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.

An online platform for the "course" abstraction

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

"Algorithms, Part I" (Summer 2012)

A screenshot of a video lecture slide. The slide is titled "Shellsort overview" and contains the following text: "Idea. Move entries more than one position at a time by h -sorting the array." Below this, it says "an h -sorted array is h interleaved sorted subsequences". There is a diagram showing a sequence of letters: L E E A M H L E P S O L T S X R. Below the letters, horizontal lines connect them to show interleaved subsequences. A small video inset in the bottom right corner shows a man with glasses speaking. At the bottom of the slide, there is a navigation bar with buttons for "Previous", "Press H for shortcuts", "Shellsort (10:48)", "Speed: 1.00x", and "Next".

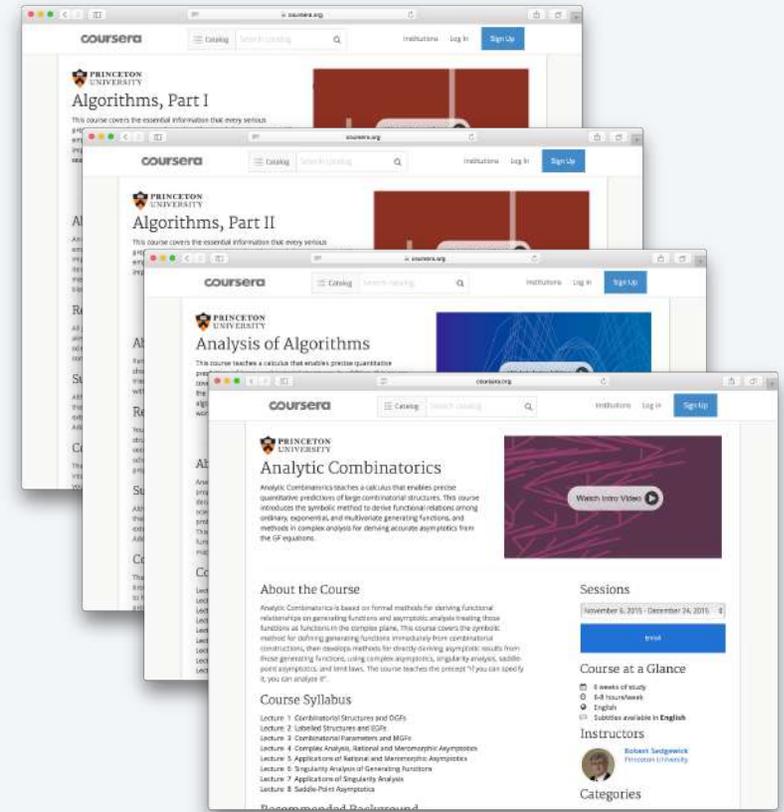
[3 years of difficult challenges omitted.]

- Production design
- Record large lectures?
- Which presentation software?
- Developing assessments
- Can we do math this way?
- Who pays?
- Who owns it?
- “You’re just a troublemaker”
- Crashing the Amazon cloud
- Builds in the presentations
- Platform issues
- Lawyers and contracts
- ...

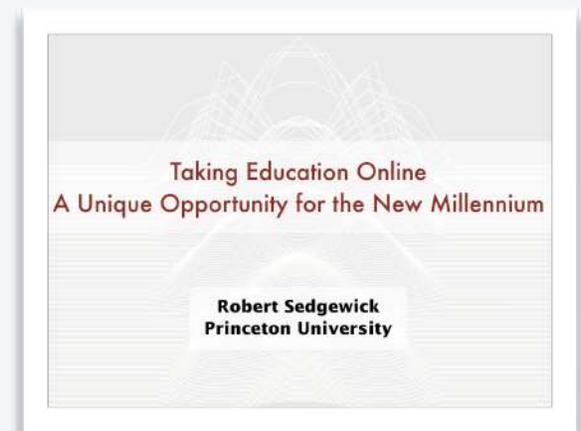
Brief summary of MOOC experience

Six courses produced in 2012–2015

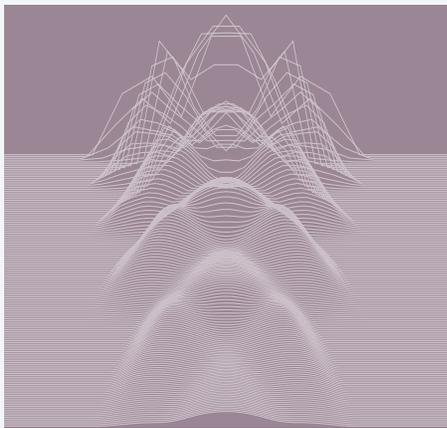
- 70+ lectures, each running 60-90 minutes
- 3000+ state-of-the-art lecture slides
- Automatic assessments in the cloud
- New platform in 2016
- Two of the top 10 courses in Coursera in both ratings and popularity
- [Stay tuned for more details]



Full story in this lecture (available online)



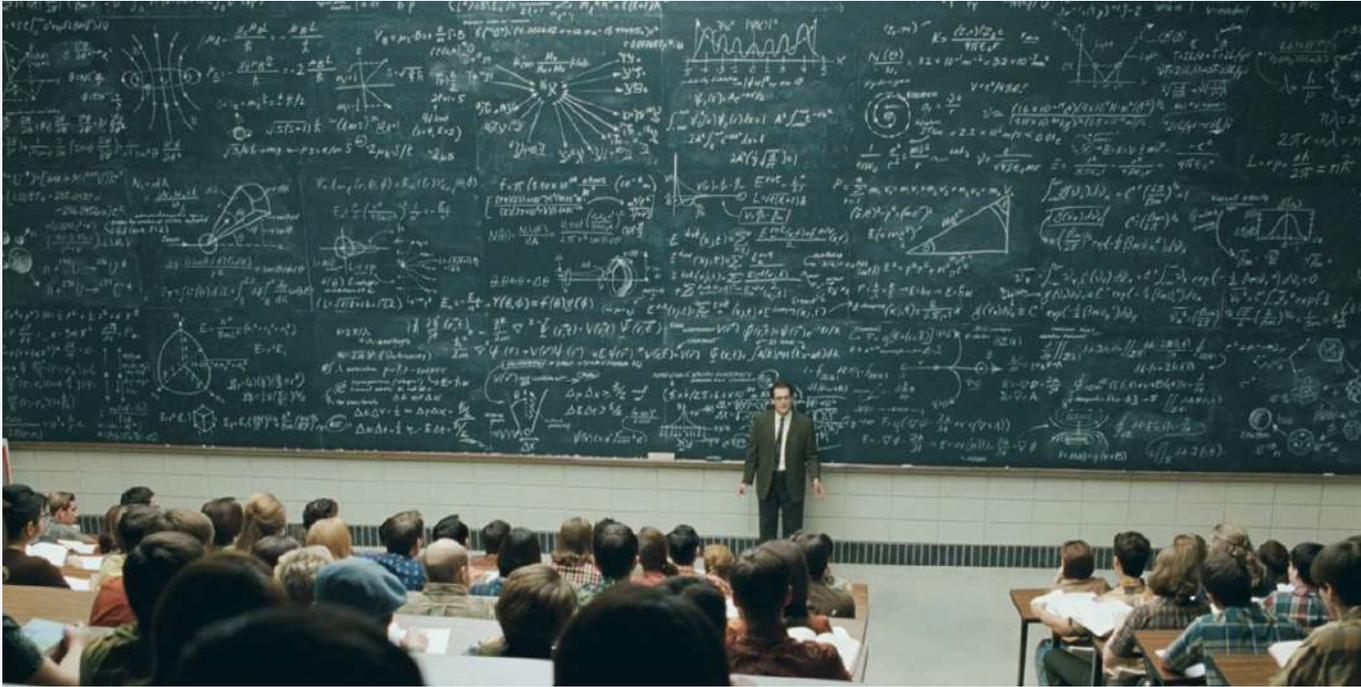
A 21st Century Model for Disseminating Knowledge



- Mission accomplished?
- Disruptive changes
- **Taking the plunge**
- A way forward
- Postscript

Sudden realization (2015)

20th Century



21st Century



Exactly how are we going to be teaching computer science *at Princeton* in the future?

RS: Hey, we *have* to use the studio-produced lectures!

Everyone else: *Why would we change our biggest and best course?*



details of debate omitted

[6 months of difficult negotiations omitted.]

- Students won't watch
- Rules won't permit it
- Will require preparation of new material
- Who will teach it
- How do we change videos?
- Video editing
- Who can watch them?
- Staff will need to reteach
- System won't support it
- Too hard to set up
- ...

Last live lecture (September 2015)

Glitches (not unusual)

- Over 90 degrees in the room
- Biggest lecture hall on campus is too small
- Students in aisles cannot see the screen
- Sound system stops working halfway through

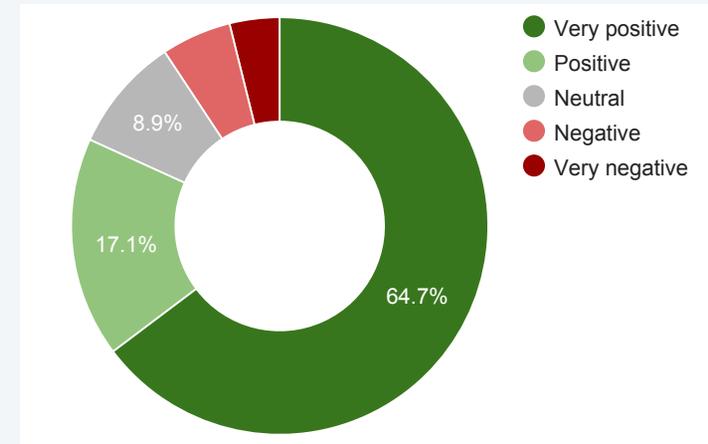


Consequence. All students motivated to move online!

An unqualified success

Q. What do you think of the online lectures?

A. 82% of responses were positive+.



Students *loved* active participation in consuming lecture content

- “Prepares me for a lifetime of active learning online.”
- “I like this system, it really lets me go at my own pace and rewatch if I need to.”
- “The video lectures are **amazing**. I believe many classes would benefit from this.”

Course staff also reaped benefits

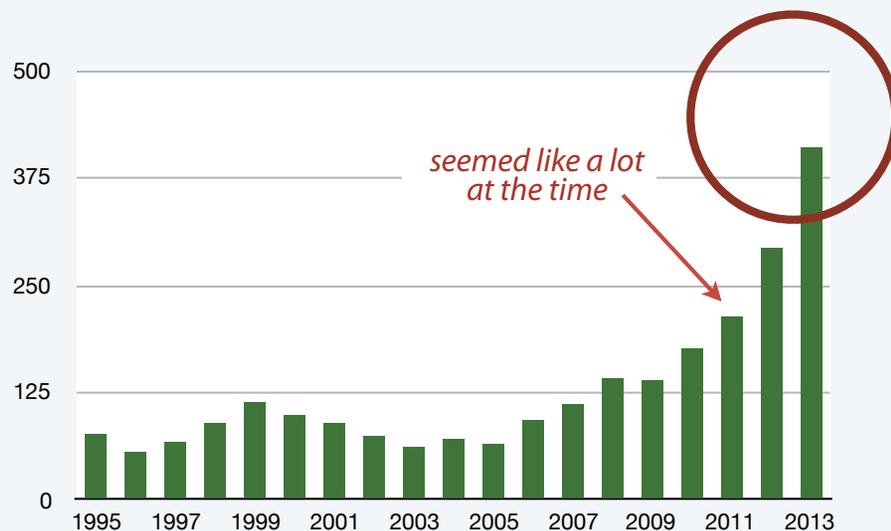
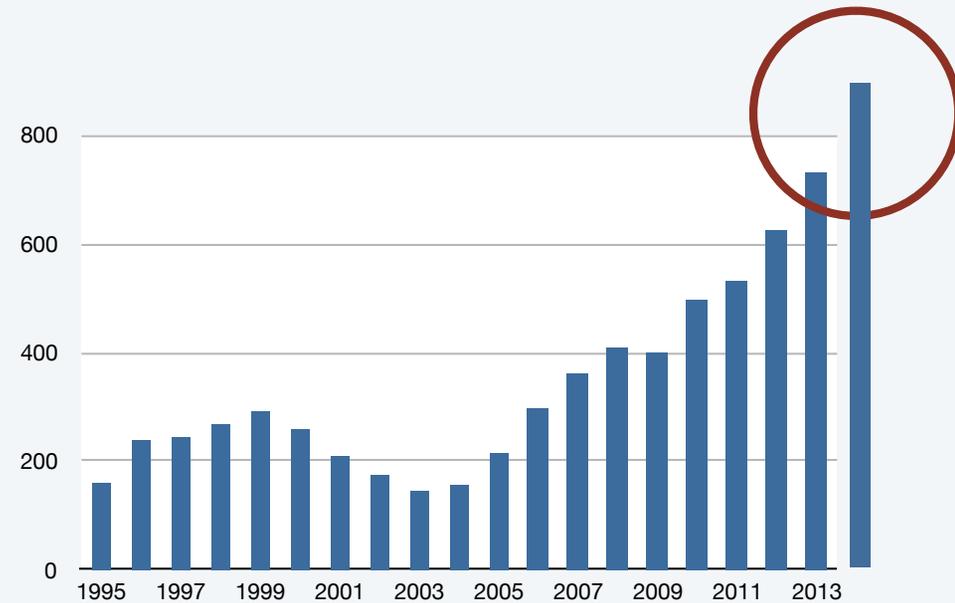
- No need to reteach lecture material in office hours.
- More time for interaction with students in small groups.
- More time for interaction in large class meetings.
- Scheduling complications virtually eliminated.

A key to success. **Not** a "flipped" class (no new material added).

Time to declare victory?

Introduction to CS enrollments

- *Triple* the height of the “bubble”
- 2/3 of all Princeton students
- Largest course at Princeton



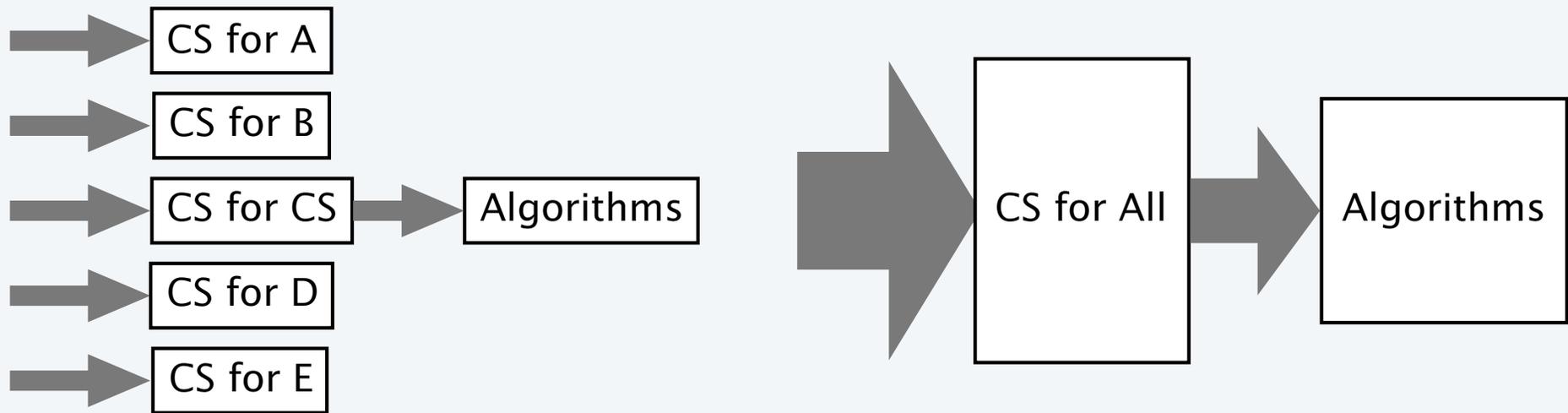
“Algorithms” enrollments

- *Four times* the height of the “bubble”.
- 40% of all Princeton students.
- 4th largest course at Princeton

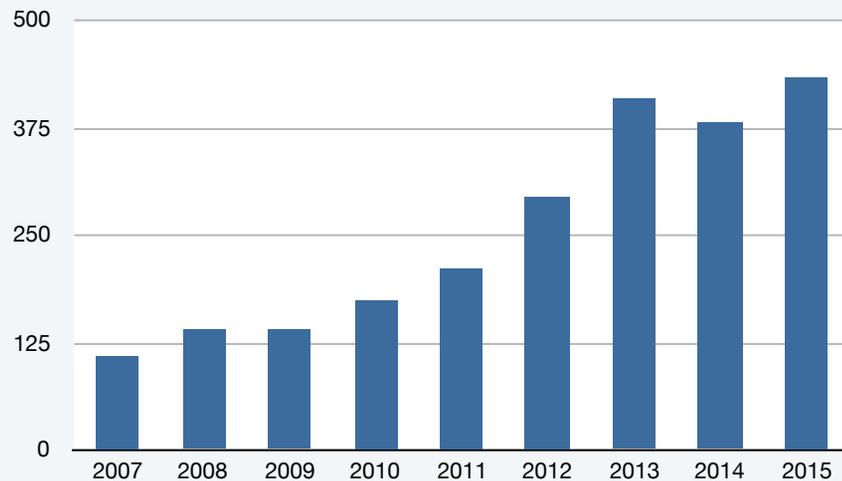
Next challenge: Attain similar percentages among *all* college students.

A bonus: Scalability plus “CS for everyone” approach

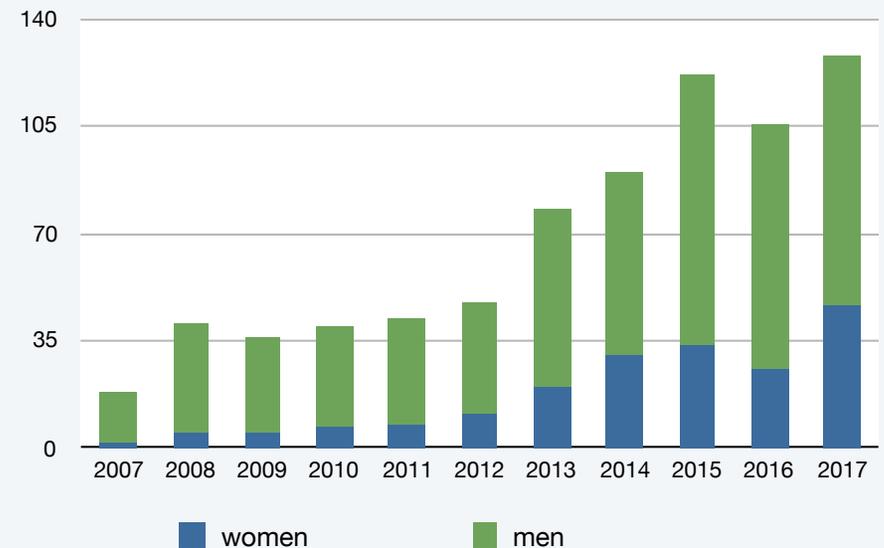
promotes *diversity* because everyone is prepared for further study in CS



Students taking a full year of CS



CS majors



Bottom Line. Nearly 40% women majors, *more than twice* the national average.

A bump in the road

Growth at this scale is *unsustainable*

- By several measures, CS is 10% of the university
- But only 5% of the faculty (!)

Three possible solutions (none likely).

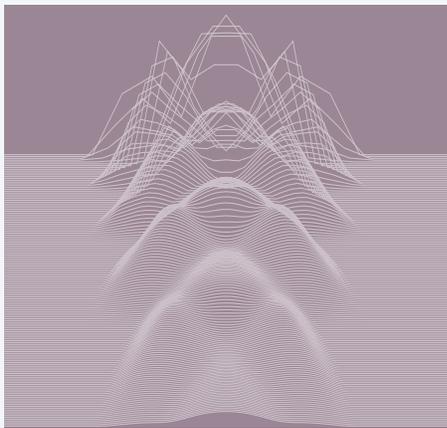
- Double the size of the CS faculty
- Double CS faculty salaries
- Limit enrollments in CS to half the demand



Lesson: A new model that scales is an *imperative for continued success*.

Good news: Such a model can *vastly extend* our reach.

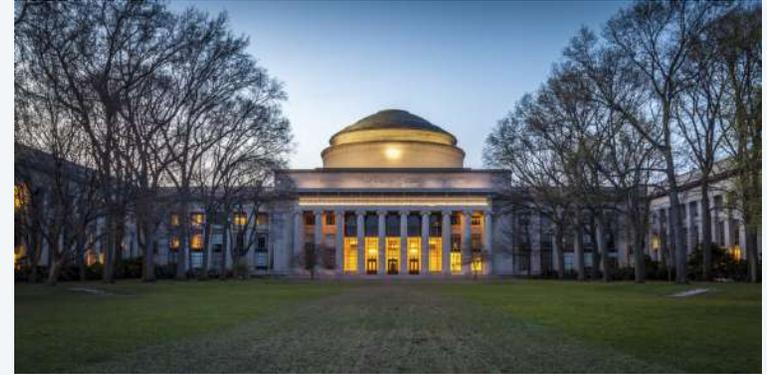
A 21st Century Model for Disseminating Knowledge



- Mission accomplished?
- Disruptive changes
- Taking the plunge
- **A way forward**
- Postscript

Purpose of the university

is to *produce and disseminate knowledge*



Holy grail for research faculty

- Excellence in teaching.
- Devotion to research.
- *Simultaneously.*
- [not going so well, lately].



A new model for teaching (this talk)

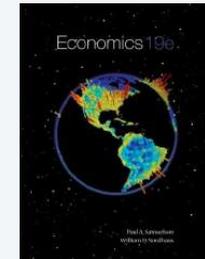
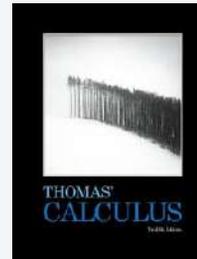
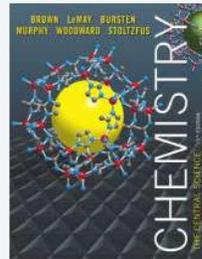
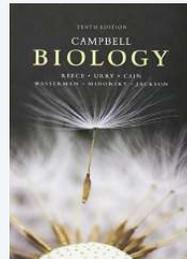
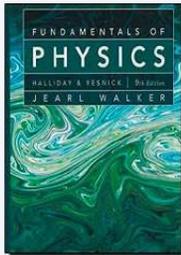
- *Replace lectures* with online videos.
- *Embrace technology* for efficiency.
- Focus on *student experience*.
- Byproduct: more time for research

20th-century textbook model

was a standard for introductory courses (in the US) *and is still widely used*

“20th century textbook” model

- “Standard” textbooks emerge after significant investment by authors/publishers.
- Distribution model: Teachers “adopt” and students buy textbooks.
- Teachers prepare and deliver lectures (perhaps using author’s slides).
- Teachers assess, grade, and certify students.



Pain points

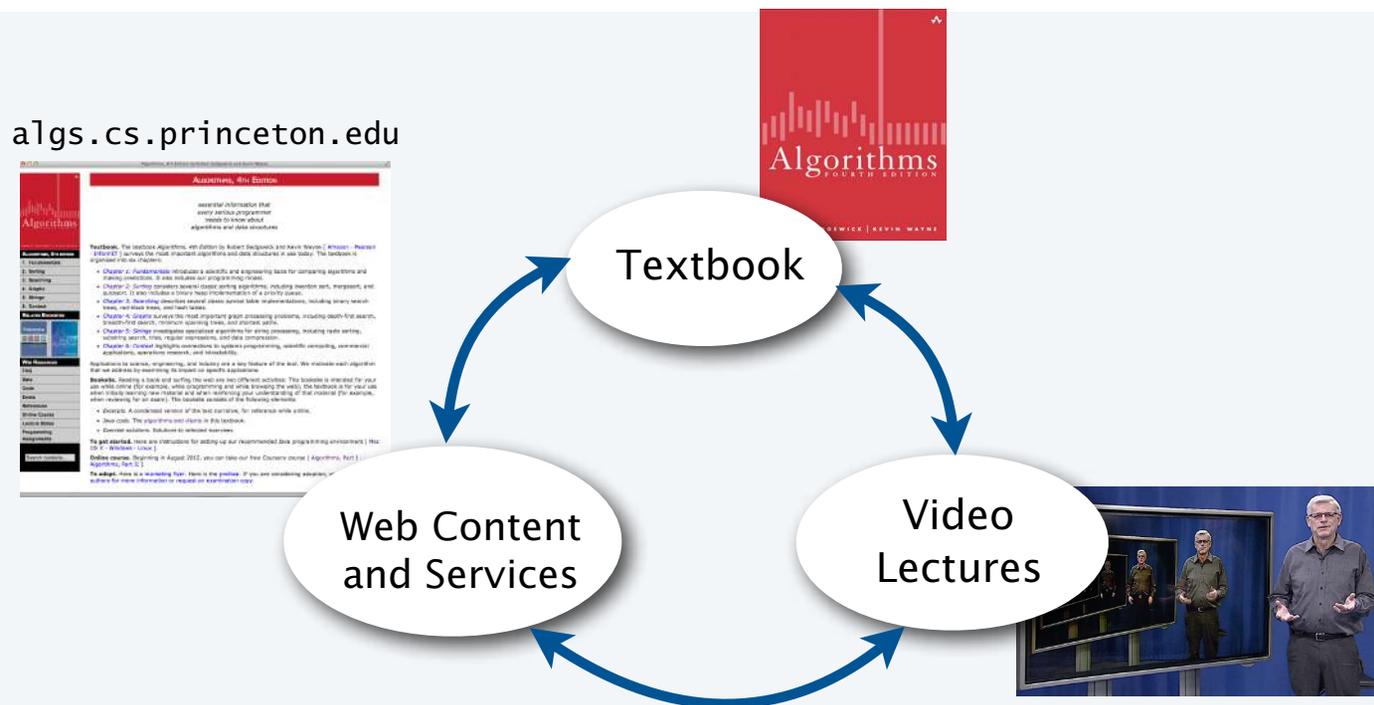
- Inefficiency of adjuncts/professors preparing and delivering “identical” lectures
- Textbook publishing imploding after move to rental model
- Most students in most lectures are *absent, bored, or lost*
- Assessment methods generally do not scale

A new model

embraces technology to integrate four abstractions that are *here to stay*

“21st century textbook” model

- Authoritative **textbook** for use to *learn* and *study* the material
- Studio-produced **video lectures** that *introduce* content and *inspire* more study
- **Web content** for use to *explore* and *interact* with the material
- **Web services** for use by teachers to *assess* and *certify* student learners



Benefits: Consistent, scalable, and flexible support of *active* teaching/learning.

Cases in point: *Computer Science, Algorithms, Analysis of Algorithms, Analytic Combinatorics*

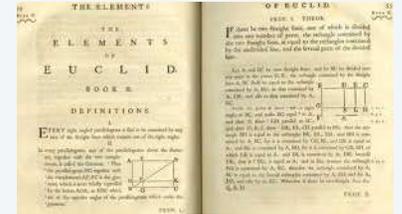
Abstraction 1: The "textbook"

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

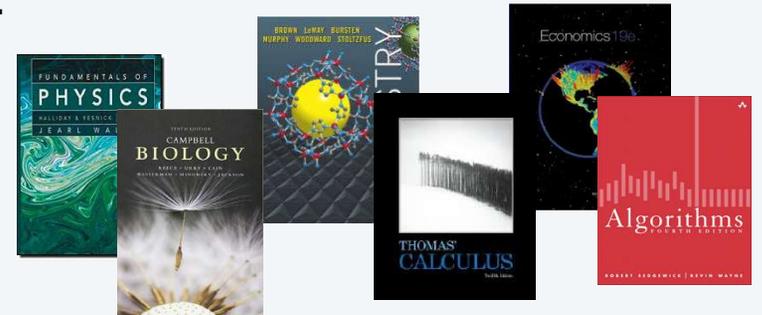


Enabled for the masses by Gutenberg.

Well-understood since the Greeks.



Still in widespread use.



Advantages

- Articulates what students can reasonably learn about a subject in a semester.
- Distills a lifetime of faculty experience for future generations.
- Provides a reference point for future studies related to the subject.

Challenges

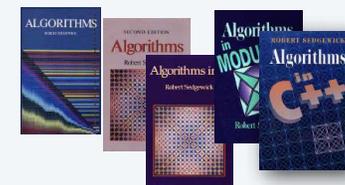
- Books need to be written by increasingly busy research-oriented professors.
- Textbook publishing industry is imploding after move to rental model.

Textbooks for CS, Algs, AofA, and AC are *alive and well*

Algorithms, Fourth Edition

Classic text for decades, 750,000+ sold.

- Completely revamped each decade
- Widely used around the world



Computer Science: An Interdisciplinary Approach

Stands with introductory books in other fields

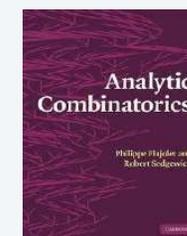
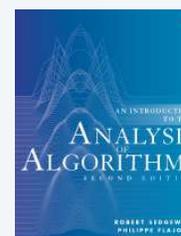
- Basis for Princeton's most popular course
- ACM *Best of Computing* Notable Book



Analysis of Algorithms and Analytic Combinatorics

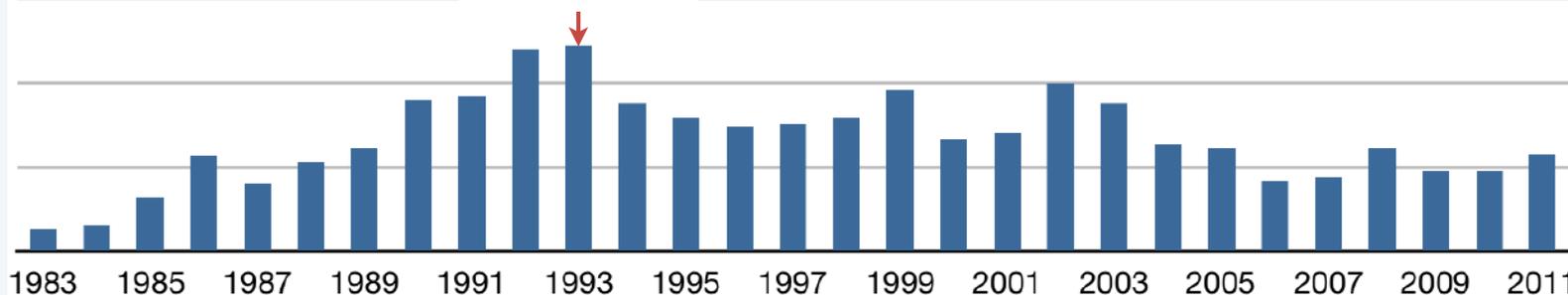
Upper-level and graduate texts

- Reflects life's work of Philippe Flajolet
- Defines a field of research



royalty income

seemed like a lot at the time



Abstraction 2: The “lecture”

has been an essential part of education for a millennium *and is about to change*

Advantages.

- Allows instructor to precisely control pace and direction.
- Stimulates development of a “community of scholars”.
- Encourages great teachers to inspire large groups of students.



U. of Bologna, founded in 1088

Disadvantages

- Requires significant time and effort for preparation.
- Duplication of effort by instructors around the world.
- *Places students in a passive role.*
- Instructor must have effective skills for production and delivery.



A university lecture in the 1350s

Source: Office of Instructional Resources, University of Illinois-Urbana Champaign.

“Lecturing is that mysterious process by means of which the contents of the note-book of the professor are transferred ... to the note-book of the student without passing through the mind of either.”

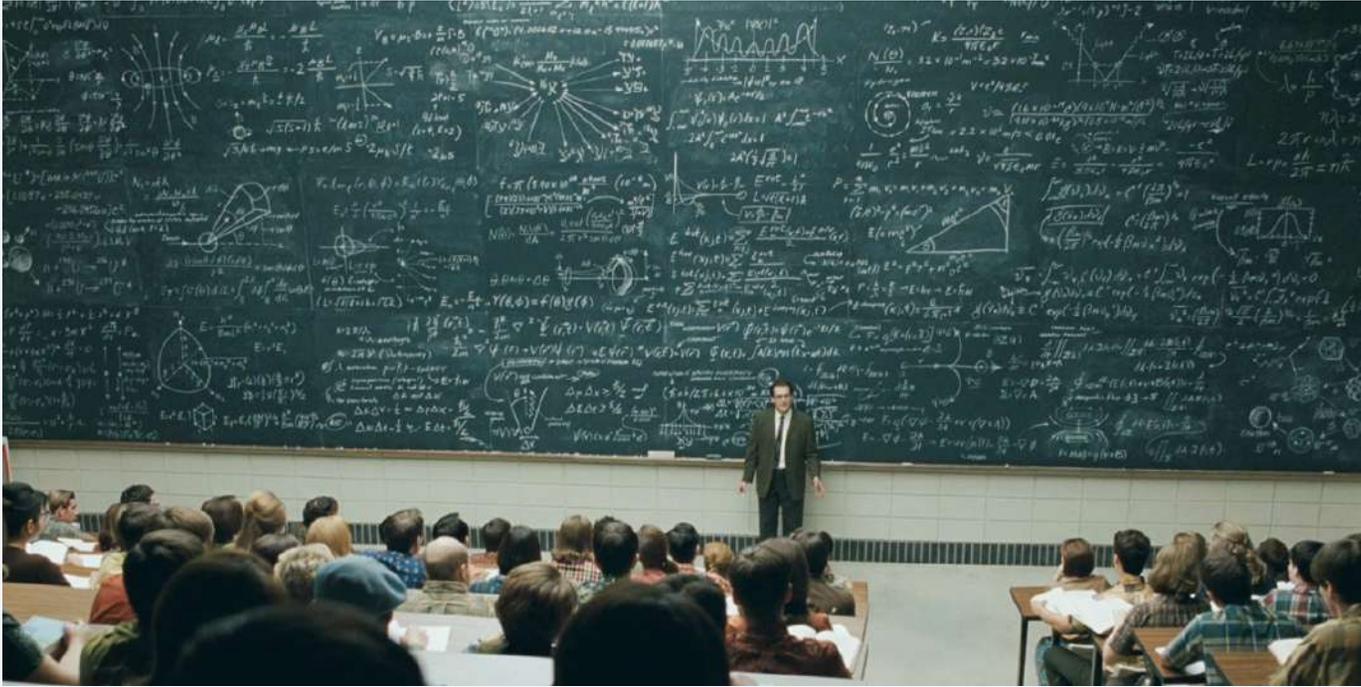
– Edwin Emery Slosson



Is there a practical alternative?

Is there a practical alternative to traditional lectures?

20th Century



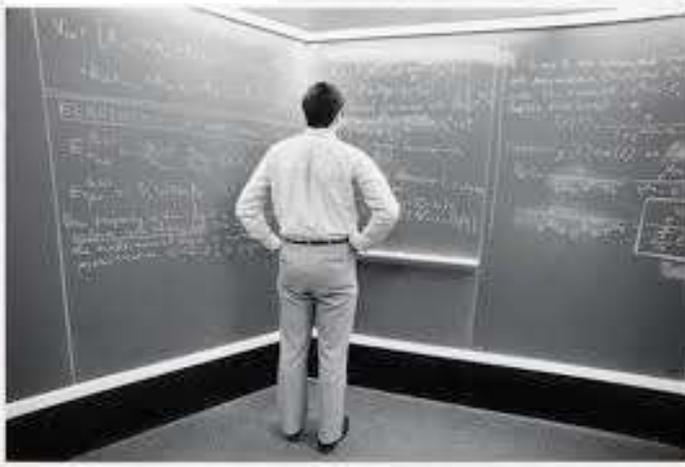
21st Century



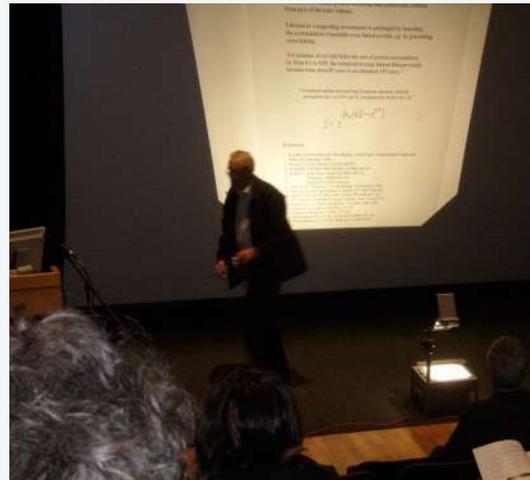
Good news: Lecture presentation materials

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

$c^r.l := gc^r.l$
 $g^r.l := gc$
 $gc^r.l := c$

"Split" any 4-nodes at 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

Connected components in mappings

| | |
|-------|--|
| Class | \mathcal{Y} , the class of cycles of Cayley trees |
| EGF | $Y(z) = \sum_{y \in \mathcal{Y}} \frac{z^{ y }}{ y !} \equiv \sum_{N \geq 0} Y_N \frac{z^N}{N!}$ |

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 $H(u) = \ln(1/(1-u))$

Lagrange Inversion Theorem (Sürmann form).
 If a CF $g(z) = \sum_{k \geq 1} g_k z^k$ satisfies the equation $z = f(g(z))$ with $f'(0) = 0$ and $f''(0) \neq 0$ then, for any function $H(u)$,
 $[z^N]H(g(z)) = \frac{1}{N} [u^{N-1}] H(u) f'(u) \left(\frac{u}{f(u)} \right)^N$

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

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

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

Presentation elements example: Analytic Combinatorics

Mathematical derivations

Cayley trees

Class C , the class of labeled rooted unordered trees

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

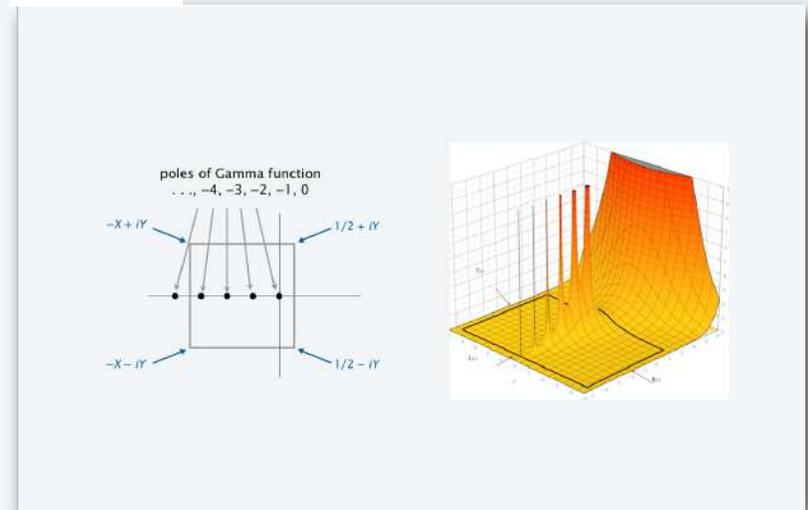
Example

6 2 1 1 2 2 5 1

52

- "Builds" control pace.
- Details to support reasoning are included.

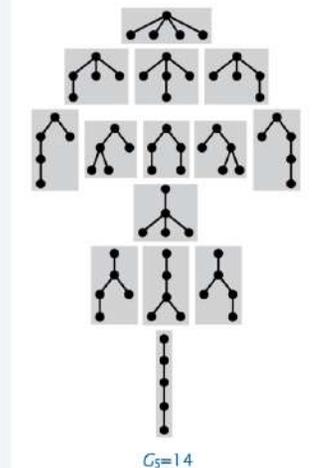
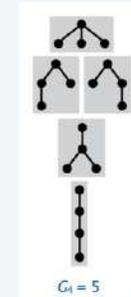
Plots



Drawings of combinatorial objects

Classic example of the symbolic method

Q. How many trees with N nodes?



"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

Good news: Studio-produced lectures

embrace technology to provide *consistent* learning experiences for students

Professional standards

- Production design
- Multiple cameras
- Video editing
- Presentation materials
- Recording sessions



By far the most time-consuming element: *content creation*

Good news: Online studio-produced lectures

transform lectures from *passive* to *active* learning experiences for students

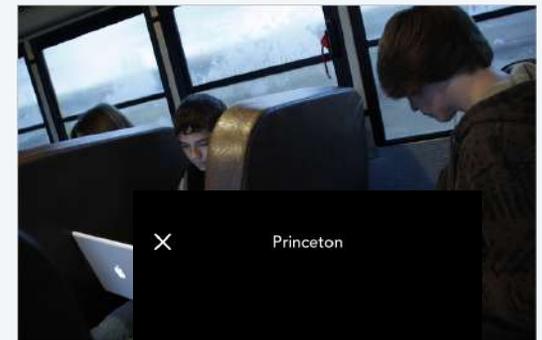
Students actively *choose their own pace*

- Typical beginners slow the pace at first.
- Typical advanced students view lectures at double speed.
- Everyone's pace varies throughout the course.
- No one is absent, lost, or bored



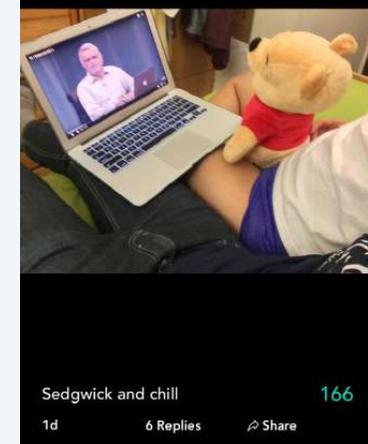
Students actively choose the time and place they learn

- “Last thing in the evening, lying in bed.”
- “First thing in the morning, in the library.”
- “On the team bus.”
- “Sedgewick and chill.”



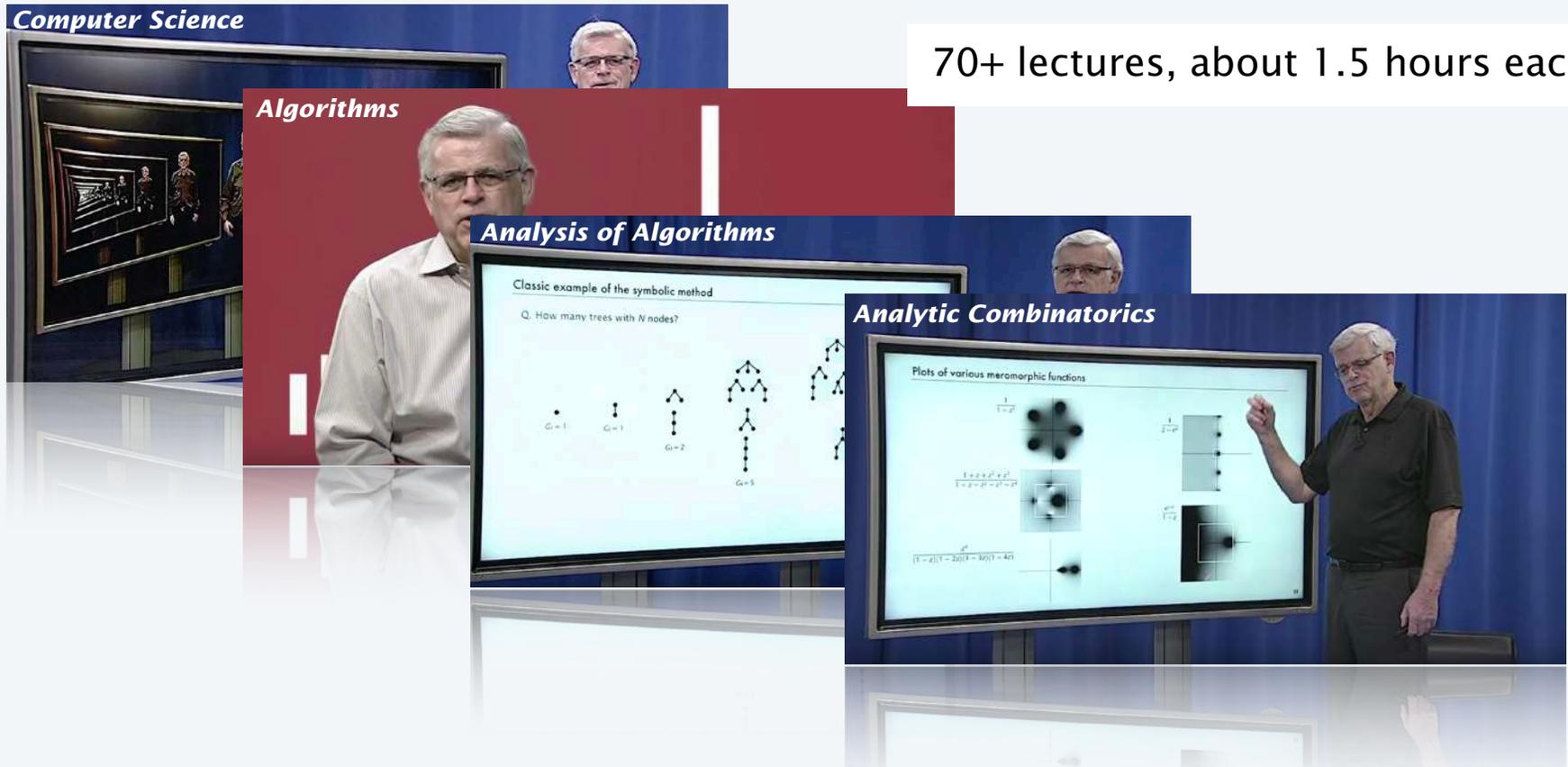
Lectures are always available for review

- Students review the material *until they understand it*.
- “Office hours” are dramatically reduced.
- Promotes diversity.
- Exam review is much less stressful.



Sedgewick and chill 166
1d 6 Replies Share

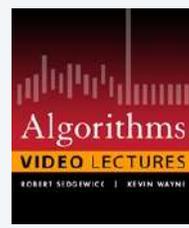
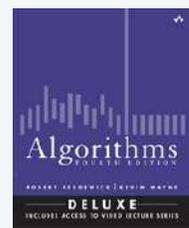
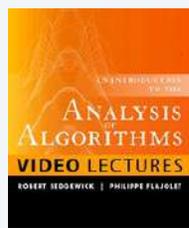
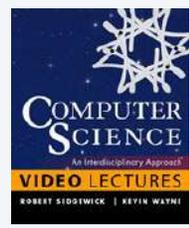
Online lectures for CS, Algs, AofA, and AC are *alive and well*



70+ lectures, about 1.5 hours each

No live lectures by RS at Princeton—online lectures used in *five* courses each year.

Widely available in various outlets, reaching over *1 million* people around the world.

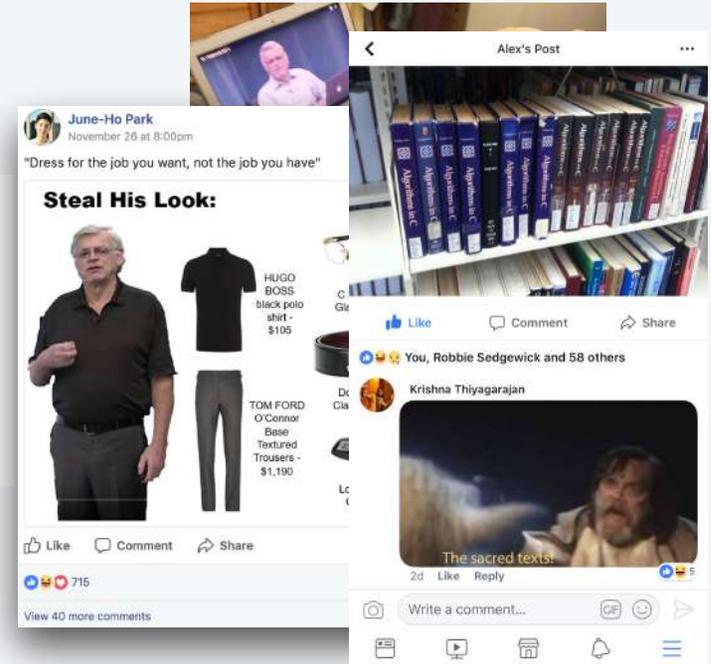


A paradox

Use of online lectures *enhances* personal interaction between faculty and students.

Videos are more personal than one might think!

- Each student spends significant time with RS.
- RS gets e-mails, waves, smiles, selfies and memes around campus and around the world.



Quality of interaction is better than with live lectures.

- Nobody is taking notes.
- Interactions are about connecting and applying ideas.
- Teachers are free to explore, knowing that basics are covered.



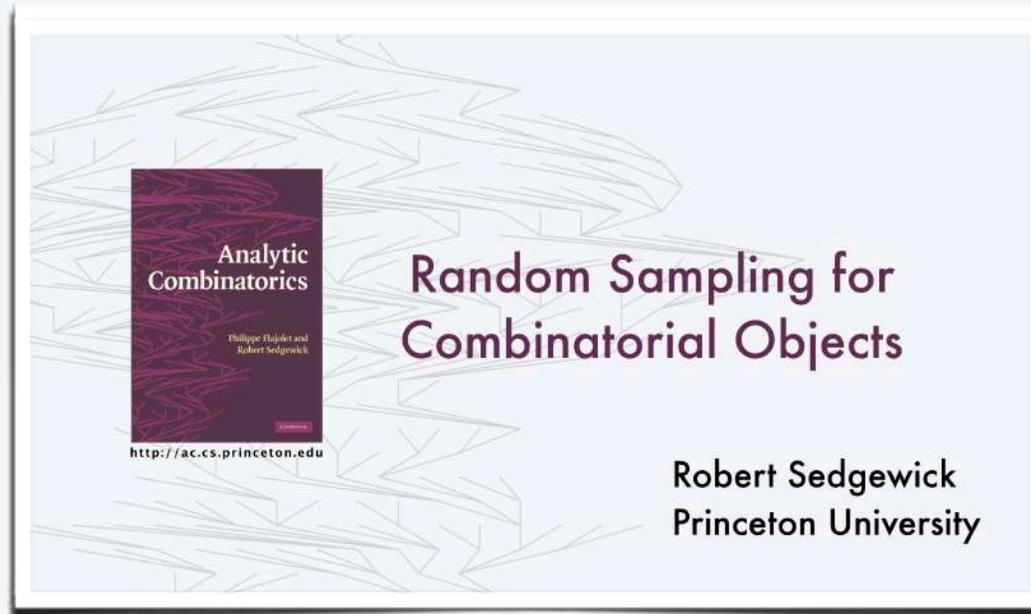
We focus on outcomes, and we care about student experiences. If we were to try to cover more material in the class meetings or try to do more of our interaction in large groups, the experience would suffer.

– David August, in response to a parent complaint

Next frontier

Q. Is it worthwhile to prepare lectures that will *never* be delivered live?

A. Of course!



Recorded May, 2018
2000+ hours of preparation
Never will be delivered live

Why?

- Opportunity to lecture to *anyone who wants to learn a topic*
- No one is lost, bored, or absent
- Wide reach justifies extra effort
- Final product is far superior to standard lectures
- When it is done, *it is done!*



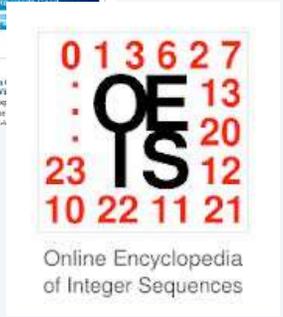
Hmmm... *Is it worthwhile to prepare live lectures?*

Abstraction 3. "Web content"

has exploded as essential in disseminating knowledge and is *here to stay*

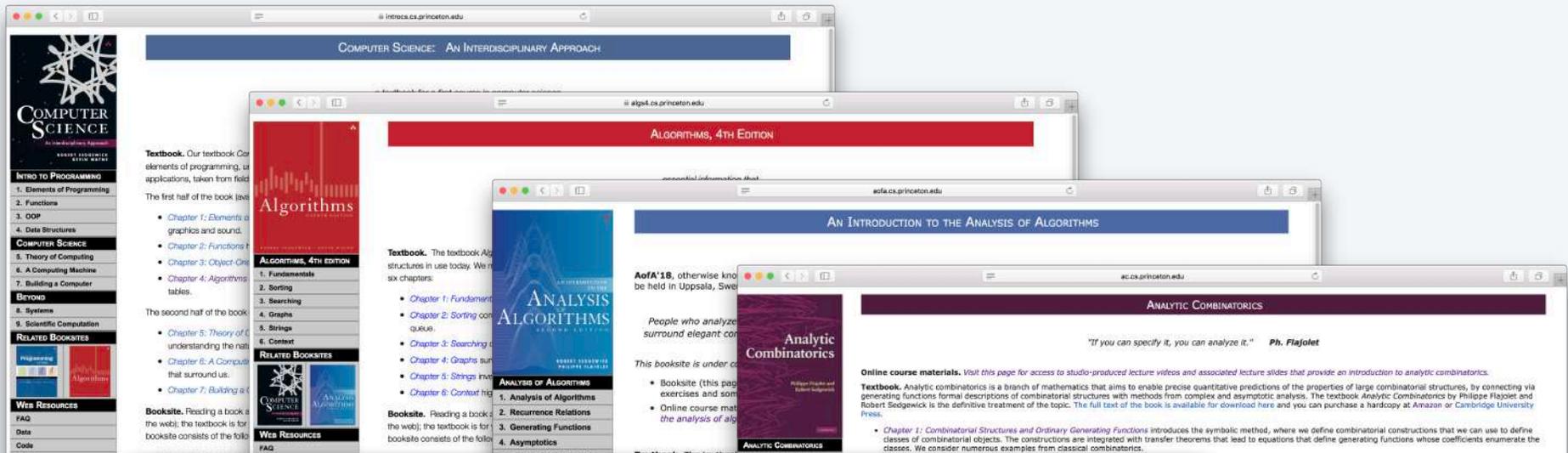


arXiv.org



- No physical constraints.
- Available to everyone.
- Always up to date (*dynamic*).
- Content types *not available* in print.

Web content for CS, Algs, AofA, and AC is *alive and well*



"Booksites" are fully integrated with the textbooks

- Web presence.
- Landing and takeoff for search.
- Code, test data, animations, exercises.
- Living documents, constantly being updated.
- For use while coding, exploring, problem solving.
- Tens of thousands of files.
- 1,000+ Java programs
- 18M+ page views in the past year
- 3M+ new users in the past year

Source: Google analytics

Abstraction 4. "Web services"

are emerging as essential in education and are *here to stay*

Relevant widely-used web services

- Tools for *creating* documents, forms, blogs.
- *Online forums* supporting Q&A and discussions.
- Web-based *testing and assessment* tools.
- Platforms for *video delivery*.



20th century: All-encompassing locally-hosted "learning management systems"

21st century: Evolving suite of homegrown and "best in class" apps running in the cloud.

Web services for CS, Algs, AofA, and AC are *alive and well*

Video delivery platforms

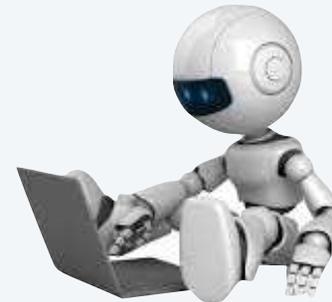


Forums and Q&A



Program assessment infrastructure

- File system/interface for student submissions.
- Dispatch mechanism to support human commentary.
- Used for many CS courses at Princeton.

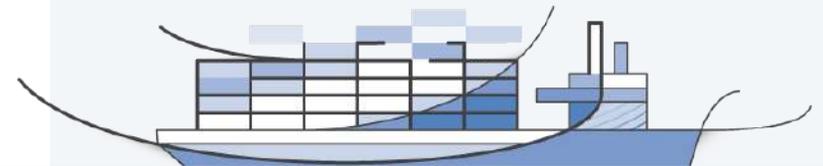


Automated program testing (stay tuned)

- Extensive fine-grained automated testing.
- Correctness (of course).
- Sophisticated performance and probabilistic tests.
- Deployed in an AWS docker container.

Quizzes and exams (stay tuned)

- Random questions drawn from templates.
- Hundreds of templates; millions of questions.
- Auto-graded for self-assessment.
- Web service in a cloud server.



Automated assessments: quizzes and exams

Example 1: Combinatorial questions

- ? Every problem in NP is also in P.
- F No problem is in both P and NP.
- T If $P = NP$ there is a polynomial-time factoring algorithm.
- ? If $P \neq NP$ there is a polynomial-time factoring algorithm.
- T There is a Turing machine that can decide whether the number of 1s on its input tape is prime.
 - .
 - .
 - .
- F The Halting Problem is NP-complete.
- T The Traveling Salesperson Problem is NP-complete.
- T There is a deterministic Turing machine that can solve every problem in NP.
- T There is a DFA that can recognize binary strings that have 1 million 0s and 1 million 1s.
- T No polynomial-time algorithm can solve the Halting Problem.

8. Computability/Intractability (5 points). For each of the computational problems below, indicate its difficulty by writing the most appropriate choice of *T* (true), *F* (false), or *?* (nobody knows) in the blank at left.

- A. _____ Every problem in NP is also in P.
- B. _____ There is a DFA that can recognize all binary palindromes.
- C. _____ There is a Turing machine that can decide whether the number of 1s on its input tape is prime.
- D. _____ No polynomial-time algorithm can solve the Halting Problem.
- E. _____ If $P = NP$ there is a polynomial-time factoring algorithm.

$$\binom{50}{5} = 2M+ \text{ questions}$$

50 facts

Automated assessments: quizzes and exams

Example 2: Data-driven questions

Q17. String sorts. The column on the left is an array of strings to be sorted. The column on the right is in sorted order. The other columns are the contents of the array at some intermediate step during one of the algorithms below. Write the letter corresponding to the correct algorithm under the corresponding column. You will need to use some letters more than once.
Hint : Do not trace code—think about algorithm invariants.

| | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| ghana | spain | aruba | aruba | aruba | chile | ghana | nepal | aruba |
| sudan | italy | benin | benin | burma | egypt | malta | sudan | benin |
| wales | ghana | burma | burma | benin | benin | china | qatar | burma |
| nepal | gabon | china | congo | china | congo | aruba | macau | chile |
| italy | libya | congo | chile | congo | burma | kenya | aruba | china |
| malta | macau | chile | china | chile | china | india | yemen | congo |
| niger | sudan | egypt | egypt | egypt | aruba | libya | niger | egypt |
| china | india | ghana | gabon | ghana | gabon | burma | wales | gabon |
| yemen | niger | gabon | ghana | gabon | ghana | zaire | congo | ghana |
| haiti | chile | haiti | haiti | haiti | haiti | chile | india | haiti |
| aruba | china | italy | kenya | italy | kenya | haiti | spain | india |
| kenya | zaire | india | spain | india | spain | nepal | benin | italy |
| spain | haiti | kenya | india | kenya | india | sudan | chile | kenya |
| india | wales | libya | libya | libya | libya | yemen | italy | libya |
| libya | malta | malta | yemen | malta | yemen | spain | burma | macau |
| burma | yemen | macau | macau | macau | macau | gabon | ghana | malta |
| macau | congo | nepal | niger | nepal | niger | benin | china | nepal |
| gabon | benin | niger | malta | niger | malta | congo | gabon | niger |
| congo | kenya | qatar | italy | qatar | italy | niger | egypt | qatar |
| benin | nepal | sudan | nepal | sudan | nepal | qatar | zaire | spain |
| egypt | burma | spain | zaire | spain | zaire | wales | malta | sudan |
| zaire | qatar | wales | wales | wales | wales | egypt | haiti | wales |
| chile | aruba | yemen | qatar | yemen | qatar | macau | kenya | yemen |
| qatar | egypt | zaire | sudan | zaire | sudan | italy | libya | zaire |

- A. *input*
- B. *LSD radix sort*
- C. *MSD radix sort*
- D. *3-way radix quicksort (no shuffle)*
- E. *sorted result*

A

E

$26^5 \times 24! = 7371780749591669477065359360000$ questions

Automated assessments: quizzes and exams

can be generated and graded in a *fully automatic* fashion.

<https://demo.quizzera.io/>

Quizzera

7KCdaeUiVUQWVgpV

Courses / Demo Course / Union Find

Quick Find

Attempts Remaining: 2

Quiz Ends in 12 days.

New Attempt

Attempts

Seed: 56642 (Provider: QuickFindExercise)

If you wish to discuss a particular question and answer in the forums or with an instructor, please post the entire question and answer, including the explanation (which contains the correct answer). Also, be sure to include the question number (located in the URL), seed, and provider which uniquely identify this question.

Note If you are unable to submit, take a screenshot, and then try refreshing the page. In most cases, refreshing will resolve any issues with the interface. Refreshing the page will keep the current attempt - it will not start a new attempt. If that doesn't work, reach out to an instructor for help. Make sure to send the screenshot of the issue!

Give the `id[]` array that results from the following sequence of 6 union operations on a set of 10 items using the quick-find algorithm.

0-9 3-5 2-8 4-3 3-7 8-7

Your answer should be a sequence of 10 integers, separated by whitespace.

Recall: our quick-find convention for the union operation `p-q` is to change `p` (and perhaps some other entries) but not `id[q]`.

Answer

To specify an array or sequence of values in an answer, separate the values in the sequence by a space. For example, if the question asks for the first ten powers of two (starting at 1), then the following answer is acceptable:

1 2 4 8 16 32 64 128 256 512

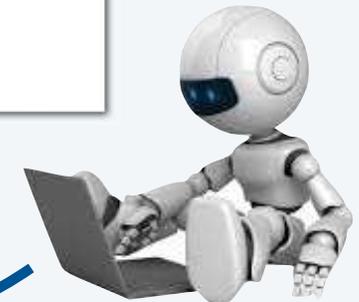
0 1 2 4 4 3 6 3 2 0

Q17. String sorts. The column on the left is an array of strings to be sorted. The column on the right is in sorted order. The other columns are the contents of the array at some intermediate step during one of the algorithms below. Write the letter corresponding to the correct algorithm under the corresponding column. You will need to use some letters more than once.

Hint: Do not trace code—think about algorithm invariants.

| | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| ghana | spain | aruba | aruba | aruba | chile | ghana | nepal | aruba |
| sudan | italy | benin | benin | burma | egypt | malta | sudan | benin |
| wales | ghana | burma | burma | benin | benin | china | qatar | burma |
| nepal | gabon | china | congo | china | congo | aruba | macau | chile |
| italy | libya | congo | chile | congo | burma | kenya | aruba | china |
| malta | macau | chile | china | china | india | yemen | congo | india |
| niger | sudan | egypt | egypt | egypt | aruba | libya | niger | egypt |
| china | india | ghana | gabon | ghana | gabon | burma | wales | gabon |
| yemen | niger | gabon | ghana | gabon | ghana | zaire | congo | ghana |
| haiti | chile | haiti | haiti | haiti | haiti | chile | india | haiti |
| aruba | china | italy | kenya | italy | kenya | haiti | spain | india |
| kenya | zaire | india | spain | india | spain | nepal | benin | italy |
| spain | haiti | kenya | india | kenya | india | sudan | chile | kenya |
| india | wales | libya | libya | libya | libya | yemen | italy | libya |
| libya | malta | malta | yemen | malta | yemen | spain | burma | macau |
| burma | yemen | macau | macau | macau | macau | gabon | ghana | malta |
| macau | congo | nepal | niger | nepal | niger | benin | china | nepal |
| gabon | benin | niger | malta | niger | malta | congo | gabon | niger |
| congo | kenya | qatar | italy | qatar | italy | niger | egypt | qatar |
| benin | nepal | sudan | nepal | sudan | nepal | qatar | zaire | spain |
| egypt | burma | spain | zaire | spain | zaire | wales | malta | sudan |
| zaire | qatar | wales | wales | wales | wales | egypt | haiti | wales |
| chile | aruba | yemen | qatar | yemen | qatar | macau | kenya | yemen |
| qatar | egypt | zaire | sudan | zaire | sudan | italy | libya | zaire |

A B C D C D B B E



7/7

Typical applications

- Self-assessments in a large class.
- Generate huge database of questions for a MOOC.
- Easy to adapt for use in a fully online class.

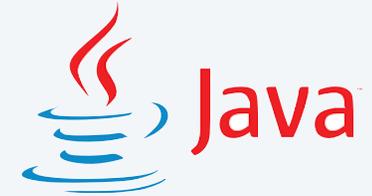
Quality and consistency of assessments are *dramatically improved* via technology.

Automated assessments: programs

can be subjected to extensive fine-grained tests and graded *automatically*.

Programs are first checked with best-in-class tools

- Every program must *compile*
- *Style checks* help develop best-practice programming habits
- Automatic *bug-finding* is essential (“because it’s easy”)



checkstyle



FindBugs
because it's easy

All assignments are based on a fully specified *API*, enabling

- *Correctness checks* (input-output pairs)
- *Timing tests* (essential in an algorithms course)
- *Memory utilization* (also essential in an algorithms course)
- *Probabilistic testing* (for randomized inputs or algorithms)

Typical applications

- Grading programs in a large class
- Grading programs in a MOOC
- Self-assessment

```
public class HelloWorld
{
    public static void main(String[] args)
    {
        // Prints "Hello, World" in the terminal window.
        System.out.println("Hello, World");
    }
}
```

100/100



Quality and consistency of assessments are *dramatically improved* via technology.

Bootstrapping

Courses produce large numbers of qualified students—why not put them to work?

Not-peer grading

- Feedback on code quality is *essential* for beginning programmers.
- Recruit students who have done well in the course to provide it. (they are *not peers*—they have another year of experience coding.)
- They can also provide *grades* to supplement automated process.
- Graders expand and reinforce their own knowledge by doing so.



Software development

- Best students are *strongly motivated* to create a killer app.
- They also seek independent research projects.
- They also understand the shortcomings of existing software.
- *Put them to work!*
- Resulting software tends to be *far better* than otherwise available.





Textbooks

Web Content and Services

Video Lectures

Computer Science: An Interdisciplinary Approach

Textbook: Our textbook Computer Science (Johnson - Flannery - Flannery) is an interdisciplinary approach to the traditional CS curriculum with Java, Haskell, and elements of programming, using an "objects in the middle" approach that emphasizes data abstraction. We review each concept by learning to implement it in applications, taken from fields ranging from materials science to genetics to astrophysics to internet commerce.

The first half of the book (available separately) see Introduction to Programming in Java is organized around four stages of learning to program:

- Chapter 1: Elements of programming introduces variables, assignment statements, built-in types of data, conditionals and loops arrays, and recursion, including graphics and sound.
- Chapter 2: Functions highlights the features of object-oriented programming that can be independently debugged, maintained, and reused.
- Chapter 3: Object-Oriented Programming
- Chapter 4: Algorithms and Data Abstraction

The second half of the book (available separately) see Introduction to Programming in Haskell is organized around four stages of learning to program:

- Chapter 5: Theory of Computation
- Chapter 6: Computing Haskell
- Chapter 7: Solving Combinatorics
- Chapter 8: Solving Combinatorics

AN INTRODUCTION TO THE ANALYSIS OF ALGORITHMS

ANALYSIS OF ALGORITHMS

ANALYTIC COMBINATORICS

Algorithms, 4th Edition

Textbook: Analytic combinatorics is a branch of mathematics that analyzes generating functions. The combinatorial structures that arise are treated in the combinatorial treatment of the book. The first half of the book covers the theory of generating functions, the combinatorial structures that arise, and the asymptotic analysis of the coefficients of generating functions. The second half of the book covers the theory of generating functions, the combinatorial structures that arise, and the asymptotic analysis of the coefficients of generating functions.

Textbook: The book Algorithms, 4th Edition by Robert Sedgwick and Kevin Wayne (Johnson - Flannery - Flannery) is the most important algorithms and data structures book to use today. It introduces each algorithm that we address by learning to implement it in applications to business, engineering, and industry. The textbook is organized into six chapters:

- Chapter 1: Fundamentals
- Chapter 2: Sorting
- Chapter 3: Searching
- Chapter 4: Graphs
- Chapter 5: Strings
- Chapter 6: Combinatorics

For students:

- Java: Here are instructions for setting up a simple Java programming environment for Mac OS X, Windows, and Linux.
- Lecture videos: The 4th edition includes professionally produced lecture videos.
- Online course: You can take our free Coursera MOOC Algorithms, Part I and Algorithms, Part II.

For instructors:

- Slides: You can request an examination copy or ask the authors for more information. Here is the [preprint](#). [ACM SIGCSE](#) also [offers](#) a course exemplar in CS2013.
- Customized: Lectures slides by Charles Leiserson are available by request. [Go to instructor page](#) to see the textbook.

Use cases: Institutions

Introductory course at a large university

- Replace live lectures with online videos
- *Teaching assistants* lead small classes
- *Professor* teaches the TAs
- *Professor and TAs* monitor online forums
- Very few *students* are absent, lost, or bored
- Focus on preparing students for success

New CS course in a third-world university

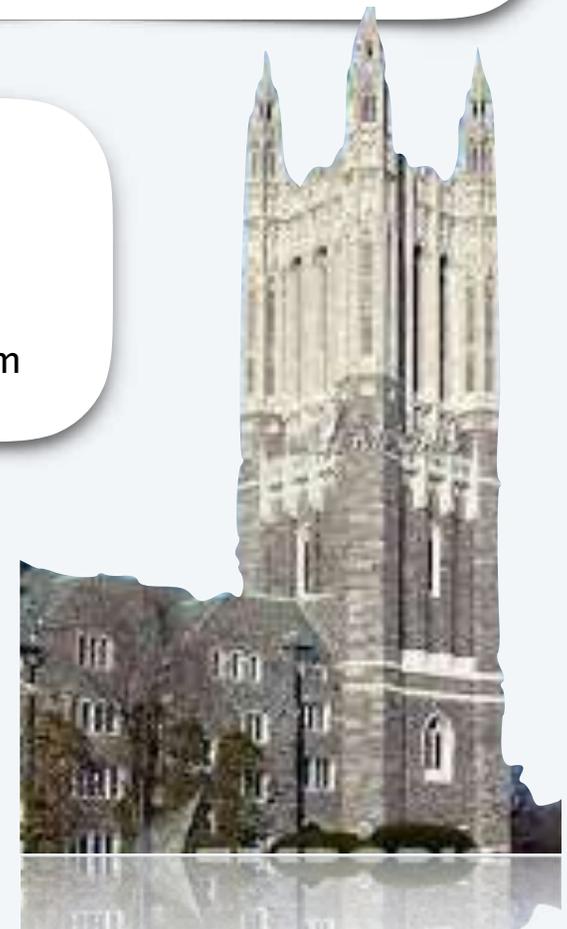
- *Students and faculty* take MOOC
- Focus on preparing students for success
- Take advantage of online content
- Bootstrap for grading and TAs after startup

Advanced course at a large university

- Replace live lectures with online videos
- *Professor* leads problem-solving sessions
- *Students* write exam questions for the next term

New CS course in a high school

- Needed: CS course like other science courses
- *Students and teacher* watch online videos
- *Teacher* makes sure everyone understands
- Teacher uses online content for assessments



Use cases: Individuals

Job seeker

- Prepares for interview with a MOOC
- Gets the job!



Third-world learner

- Limited educational opportunities
- Free MOOCs available via smartphone

College student

- Lost or bored during lecture
- Watches online video in class



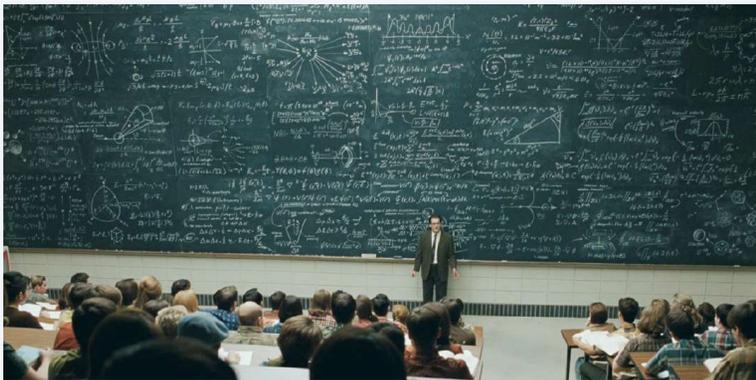
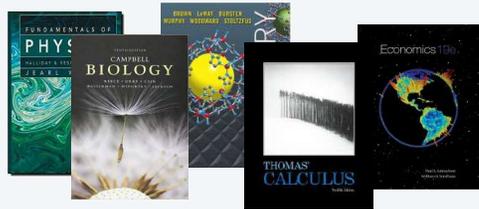
High school student

- Not much to do after college acceptance
- Takes MOOC with friends in the summer
- Places out of college course

Mid-career professional

- Was absent, lost or bored while in college
- Needs access to specific knowledge for job
- No interest in assessments or certificates

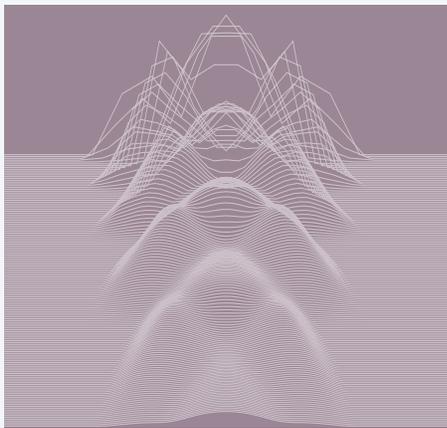
20th century



21st century



A 21st Century Model for Disseminating Knowledge



- Mission accomplished?
- Disruptive changes
- Taking the plunge
- A way forward
- **Postscript**

Questions answered (faculty)



Q. Isn't developing an online course time-consuming and difficult?

A. Yes! 50-100 hours of preparation per lecture.

A. Yes! My workflow requires skill in Unix, emacs, Mac OS X, DropBox, Illustrator, Indesign, Java, C, TeX, video capture, KeyNote, Acrobat, HTML, MathJax, Mathematica, PostScript, and a dozen other tools I can't even name.

A. It's less time-consuming than writing a book.

A. Not everyone needs to do it, so you have more time for research.

Q. I think my administration will fire *me* and use *your* lectures.

A. I'm wondering about that myself!

A. Cut costs by firing teachers? I think not.

A. They've already "fired" your successors by not hiring enough CS faculty.

A. Your students need you to help them succeed and certify that they did so.

A. Does it make sense for hundreds of professors around the world to be preparing and delivering lectures on Quicksort, hashing, and a dozen other topics every semester?

A. Are you planning to spend 50-100 hours of prep time?

A. Your students are *already* watching my lectures.

Questions answered (administrators)



- Q. Why aren't more faculty developing online courses?
A. It's difficult! And you are evaluating them on research, not this.
- Q. Please do a scientific study on online vs. traditional lectures.
A. Sure. Please provide the % of students attending traditional.
(I know the answer for online.)
- A. Did you do a study when your faculty switched to PowerPoints?

Q. What can we do to improve things?

A. *"Respond with urgency" to the National Academies report on CS education.*

A. Develop a *teaching class* of professors who can teach blended classes.

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

A. Invest in *research* at the interface of education and technology.

A. Attract and provide resources to the best and brightest professors in the field.

A. Develop *academic leadership* for charting the future.

A. *Embrace technology.*

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 this at the scale you are investing in the library.*

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

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.*”



What happened to the tsunami?

I think the bloom is now off the rose, and now is going to be the time when some really hard-nosed thinking has to be done about the true value of these online courses.

Shirley Tilghman, 2013



Stumbling blocks

- *Institutions* are trying to take control (and failing).
- Content creation is the province of *individuals*.
- Bad business models, created prematurely.

Result: Plenty of lost opportunities.

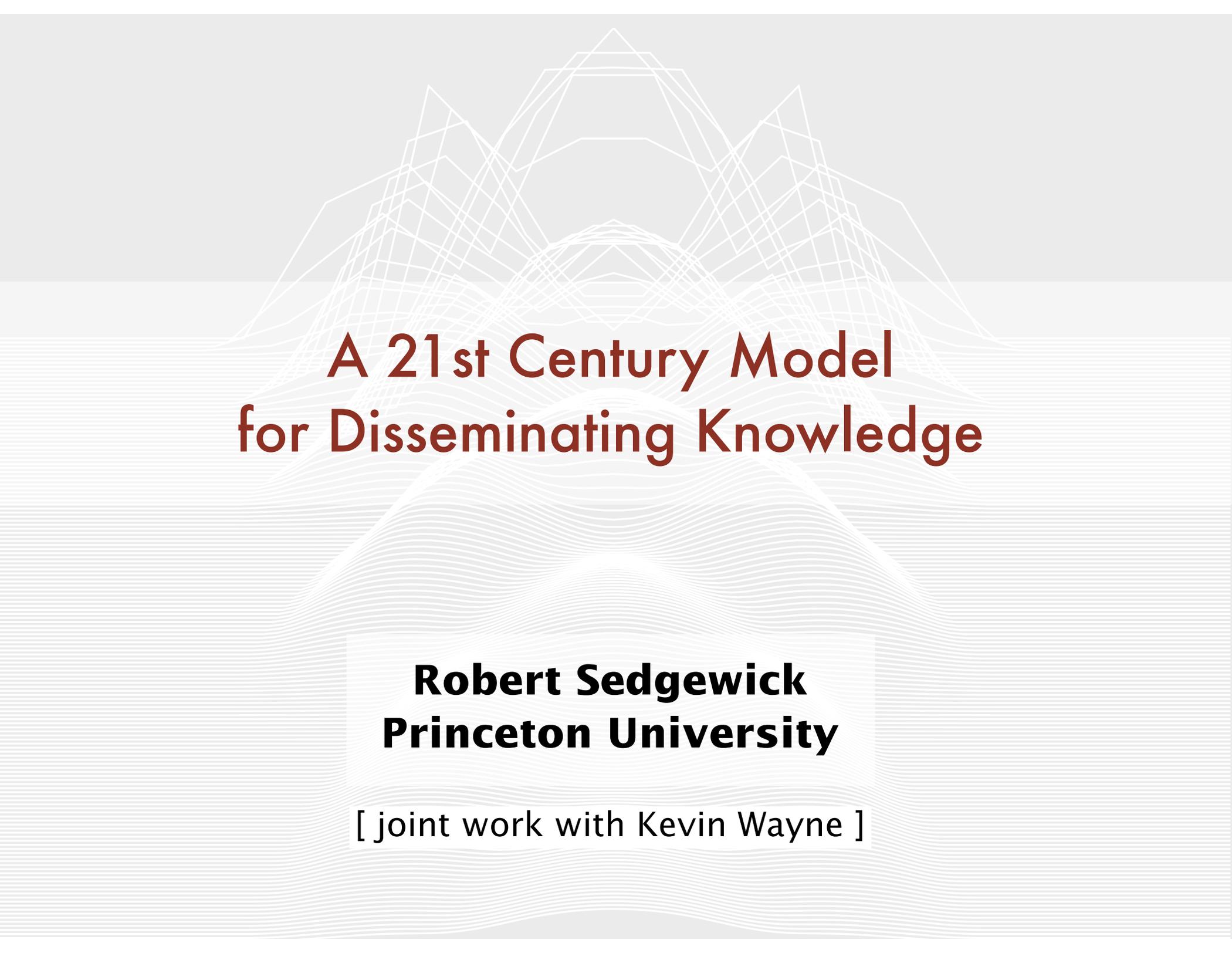


Still to emerge: 21st-century models connecting content creators and consumers.

RS: Looks like a tsunami to me!

- Tens of thousands of pages of online content
- 100+ hours of lecture videos.
- Reaching millions of individuals.
- 1990s: Lucky to be able to teach my own children.
- 2030s: Will be teaching my own *grandchildren*.





A 21st Century Model for Disseminating Knowledge

Robert Sedgewick
Princeton University

[joint work with Kevin Wayne]