A 21st Century Model for Disseminating Knowledge

Robert Sedgewick
Princeton University

[joint work with Kevin Wayne]
A 21st Century Model for Disseminating Knowledge

- Mission accomplished?
- Disruptive changes
- Taking the plunge
- A way forward
- Postscript
Fundamental challenge for teaching CS (1975-present)

*Standard textbooks* have been the norm in most fields in the US for decades.
Central thesis (RS, 1975, 1992)

All college students need courses in computer science and algorithms

Computer science embraces a significant body of knowledge that is
• intellectually challenging
• pervasive in modern life
• critical to modern science and engineering

Anyone can learn the importance of
• modern programming models
• the scientific method in understanding program behavior
• algorithms and data structures
• fundamental precepts of computer science
• 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.
Basic approach

General landscape in 1992 (still true, somewhat)
- All students get basic education in science (from HS)
- Many scientists view computer science and algorithms as just tools.
- Many computer scientists take a largely abstract view of the field.

Idea: Leverage students’ basic knowledge of science to teach CS.

1970s: Anyone using the computer needs CS/algorithms.
1990s: Future cubicle-dwellers need CS/algorithms.
2010s: Everyone needs CS/algorithms.

To address the situation:
- Teach the fundamentals to everyone.
- Do it as early as possible.
[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
- ...
Sedgewick & Wayne courses at Princeton (2011)

**Computer Science**
- For first-year students.
- Includes programming in Java through data abstraction.
- Teaches fundamental ideas in the field including theory of computation and architecture.
- Stands alongside intro courses in economics, physics, biology, and other disciplines.

**Algorithms**
- For first-year students.
- Full coverage including graphs and strings.
- Emphasize on scientific method for studying performance.
- Widely adopted worldwide.
- Under development since 1974.

*Interdisciplinary* approach that embraces, leverages and supports other disciplines.
**2011: Time to declare victory?**

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

---

**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.

---

**Q. (2011) When will enrollments start to decline?**
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:
- 40+% of all college students in an intro CS course
- 25+% of all college students in an algorithms course

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

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)

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

Is there room for a good textbook?
Will free web resources prevail?
Disruptive change III
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...

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

With apologies to our actual administrators

Andrew Ng and Daphne Koller

Coursera

An online platform for the "course" abstraction

"Algorithms, Part I" (Summer 2012)
[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

Facts and figures
• Six courses produced, four already deployed.
• 70+ lectures, each running 60-90 minutes.
• 3000+ state-of-the-art lecture slides.
• Over 1 million people reached.

Distribution model is evolving (stay tuned)
• Intro courses offered as Coursera MOOCs.
• Advanced courses freely available (2017)
• Each course has an associated textbook.
• Lecture videos also bundled with the textbooks.
• Each textbook has associated web content.
A 21st Century Model for Disseminating Knowledge

- Mission accomplished?
- Disruptive changes
- Taking the plunge
- A way forward
- Postscript
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?
[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!
Q. What do you think of the online lectures?

A. 82% of responses were positive+.

**An unqualified success**

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.
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.
Scalability plus “CS for everyone” approach promotes *diversity* because *everyone* is prepared for further study in CS.

Bottom Line. Nearly 40% women majors, *more than twice* the national average.
A 21st Century Model for Disseminating Knowledge

- Mission accomplished?
- Disruptive changes
- Taking the plunge
- A way forward
- Postscript
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.
• Passive lecture experience has become unsustainable.
• Assessment efforts generally do not scale.
21st-century textbook 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.
Abstraction 1: The "textbook"

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.

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.
Good news: technology is empowering individuals to be more productive than ever before.

even busy research-oriented professors
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.

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.

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

“A 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

State of the art presentations

"PowerPoints" are evolving to new standard of excellence

This is a horse.
Presentation elements example: Analytic Combinatorics

Mathematical derivations

<table>
<thead>
<tr>
<th>Class</th>
<th>$C$, the class of labeled rooted unordered trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>EGF</td>
<td>$C(z) = \sum_{</td>
</tr>
</tbody>
</table>

Example:

```
   2  1  7  6
  3  4
  8  1
```

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

Plots

Drawings of combinatorial objects

"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

Classic example of the symbolic method

Q. How many trees with $N$ nodes?
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.

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.
Abstraction 3. "Web content" has exploded as essential in disseminating knowledge and is *here to stay*

- No physical constraints.
- Available to everyone.
- Always up to date (*dynamic*).
- Content types *not available* in print.
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.
A 21st Century Model for Disseminating Knowledge

- Mission accomplished?
- Disruptive changes
- Taking the plunge
- A way forward
- Case in point
- Postscript
Case in point: Algorithms, Fourth Edition is a 21st-century textbook that is already fully deployed.

**An integrated, effective, and scalable model**
- **Textbook** under continuous development and a best-seller since 1982.
- **Web content** under development and widely used since mid 2000s.
- Innovative **web services** are continuing to evolve
- **Studio-produced videos** in widespread use since 2012.
Algorithms textbook

**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.

<table>
<thead>
<tr>
<th>edition</th>
<th>goal for code: clear, readable and</th>
</tr>
</thead>
<tbody>
<tr>
<td>1970s</td>
<td>1st</td>
</tr>
<tr>
<td>1980s</td>
<td>2nd</td>
</tr>
<tr>
<td>1990s</td>
<td>3rd</td>
</tr>
<tr>
<td>2010s</td>
<td>4th</td>
</tr>
</tbody>
</table>
Algorithms web content

http://algs4.cs.princeton.edu/

Developed by Kevin Wayne since the mid-2000s

Fully integrated with the textbook
• Web presence.
• Landing and takeoff for search.
• Code, test data, animations.
• A living document.
• For use while coding, exploring.
• 10,000+ files.
• 1,000+ Java programs
• 6M+ page views in the past year
• 1M+ unique users in the past year
Algorithms online lecture videos

Fully integrated with textbook
- A “top 10 MOOC of all time”.
- 24 lectures, about 1.5 hours each.
- Also distributed by Pearson/InformIT.
- Widely used around the world.
- Have reached 1M+ people.
Algorithms web services

Video delivery platform

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

Forums and Q&A

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.
Example 1: Combinatorial questions

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.

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.

\[ \binom{50}{5} = 2M+ \text{ questions} \]
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.

<table>
<thead>
<tr>
<th>ghana</th>
<th>spain</th>
<th>aruba</th>
<th>aruba</th>
<th>aruba</th>
<th>chile</th>
<th>ghana</th>
<th>nepal</th>
<th>aruba</th>
</tr>
</thead>
<tbody>
<tr>
<td>sudan</td>
<td>italy</td>
<td>benin</td>
<td>benin</td>
<td>burma</td>
<td>egypt</td>
<td>malta</td>
<td>sudan</td>
<td>benin</td>
</tr>
<tr>
<td>wales</td>
<td>ghana</td>
<td>burma</td>
<td>burma</td>
<td>benin</td>
<td>benin</td>
<td>china</td>
<td>qatar</td>
<td>burma</td>
</tr>
<tr>
<td>nepal</td>
<td>gabon</td>
<td>china</td>
<td>congo</td>
<td>china</td>
<td>congo</td>
<td>aruba</td>
<td>macau</td>
<td>chile</td>
</tr>
<tr>
<td>italy</td>
<td>libya</td>
<td>congo</td>
<td>chile</td>
<td>congo</td>
<td>burma</td>
<td>kenya</td>
<td>aruba</td>
<td>china</td>
</tr>
<tr>
<td>malta</td>
<td>macau</td>
<td>chile</td>
<td>chile</td>
<td>china</td>
<td>india</td>
<td>yemen</td>
<td>cong</td>
<td>yemen</td>
</tr>
<tr>
<td>niger</td>
<td>sudan</td>
<td>egypt</td>
<td>egypt</td>
<td>egypt</td>
<td>aruba</td>
<td>libya</td>
<td>niger</td>
<td>egypt</td>
</tr>
<tr>
<td>china</td>
<td>india</td>
<td>ghana</td>
<td>gabon</td>
<td>ghana</td>
<td>gabon</td>
<td>burma</td>
<td>wales</td>
<td>gabon</td>
</tr>
<tr>
<td>yemen</td>
<td>niger</td>
<td>gabon</td>
<td>ghana</td>
<td>gabon</td>
<td>ghana</td>
<td>zaire</td>
<td>cong</td>
<td>ghana</td>
</tr>
<tr>
<td>ghana</td>
<td>chile</td>
<td>haiti</td>
<td>haiti</td>
<td>haiti</td>
<td>haiti</td>
<td>chile</td>
<td>india</td>
<td>haiti</td>
</tr>
<tr>
<td>aruba</td>
<td>china</td>
<td>italy</td>
<td>kenya</td>
<td>italy</td>
<td>kenya</td>
<td>haiti</td>
<td>spain</td>
<td>india</td>
</tr>
<tr>
<td>kenya</td>
<td>zaire</td>
<td>india</td>
<td>spain</td>
<td>india</td>
<td>spain</td>
<td>nepal</td>
<td>benin</td>
<td>italy</td>
</tr>
<tr>
<td>spain</td>
<td>haiti</td>
<td>kenya</td>
<td>india</td>
<td>kenya</td>
<td>india</td>
<td>sudan</td>
<td>chile</td>
<td>kenya</td>
</tr>
<tr>
<td>india</td>
<td>wales</td>
<td>libya</td>
<td>libya</td>
<td>libya</td>
<td>libya</td>
<td>yemen</td>
<td>italy</td>
<td>libya</td>
</tr>
<tr>
<td>libya</td>
<td>malta</td>
<td>malta</td>
<td>malta</td>
<td>malta</td>
<td>malta</td>
<td>yemen</td>
<td>spain</td>
<td>burma</td>
</tr>
<tr>
<td>burma</td>
<td>yemen</td>
<td>macau</td>
<td>macau</td>
<td>macau</td>
<td>macau</td>
<td>gabon</td>
<td>ghana</td>
<td>malta</td>
</tr>
<tr>
<td>macau</td>
<td>congo</td>
<td>nepal</td>
<td>nepal</td>
<td>nepal</td>
<td>nepal</td>
<td>benin</td>
<td>china</td>
<td>nepal</td>
</tr>
<tr>
<td>gabon</td>
<td>benin</td>
<td>niger</td>
<td>malta</td>
<td>niger</td>
<td>malta</td>
<td>cong</td>
<td>gabon</td>
<td>malta</td>
</tr>
<tr>
<td>congo</td>
<td>kenya</td>
<td>qatar</td>
<td>italy</td>
<td>qatar</td>
<td>italy</td>
<td>niger</td>
<td>egypt</td>
<td>qatar</td>
</tr>
<tr>
<td>benin</td>
<td>nepal</td>
<td>sudan</td>
<td>nepal</td>
<td>sudan</td>
<td>nepal</td>
<td>qatar</td>
<td>zaire</td>
<td>spain</td>
</tr>
<tr>
<td>egyp</td>
<td>burma</td>
<td>spain</td>
<td>zaire</td>
<td>spain</td>
<td>zaire</td>
<td>wales</td>
<td>malta</td>
<td>sudan</td>
</tr>
<tr>
<td>zaire</td>
<td>qatar</td>
<td>wales</td>
<td>wales</td>
<td>wales</td>
<td>egypt</td>
<td>haiti</td>
<td>wales</td>
<td>wales</td>
</tr>
<tr>
<td>chile</td>
<td>aruba</td>
<td>yemen</td>
<td>qatar</td>
<td>yemen</td>
<td>qatar</td>
<td>macau</td>
<td>kenya</td>
<td>yemen</td>
</tr>
<tr>
<td>qatar</td>
<td>egypt</td>
<td>zaire</td>
<td>sudan</td>
<td>zaire</td>
<td>sudan</td>
<td>italy</td>
<td>libya</td>
<td>zaire</td>
</tr>
</tbody>
</table>

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

\[26^5 \times 24! = 7371780749591669477065359360000 \text{ questions}\]
Algorithms automated assessments: quizzes and exams can be generated and graded in a fully automatic fashion.

https://demo.quizzera.io/

Typical applications

- Self-assessments in a large flipped 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.
Algorithms 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”).

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 flipped class.
• Grading programs in a MOOC.
• Easy to adapt for use in an online class.

Quality and consistency of assessments are dramatically improved via technology.
IF YOU DON'T TURN IN AT LEAST ONE HOMEWORK ASSIGNMENT, YOU'LL FAIL THIS CLASS.

YEAH. BUT IF I CAN FAIL THIS CLASS, THE GRADES ON MY REPORT CARD WILL BE IN ALPHABETICAL ORDER!
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 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.
20th century

Textbook

Web Content and Services

Video Lectures

21st century
New in 2016: Computer Science

**Computer Science**
- Web content under development since 2000.
  - 2M+ unique users and 8M+ page views in the past year.
- Textbook *published June 2016*.

Also available: *Analysis of Algorithms* and *Analytic Combinatorics* (free).
A 21st Century Model for Disseminating Knowledge

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

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. 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. Embrace technology.
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.

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.

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 ]