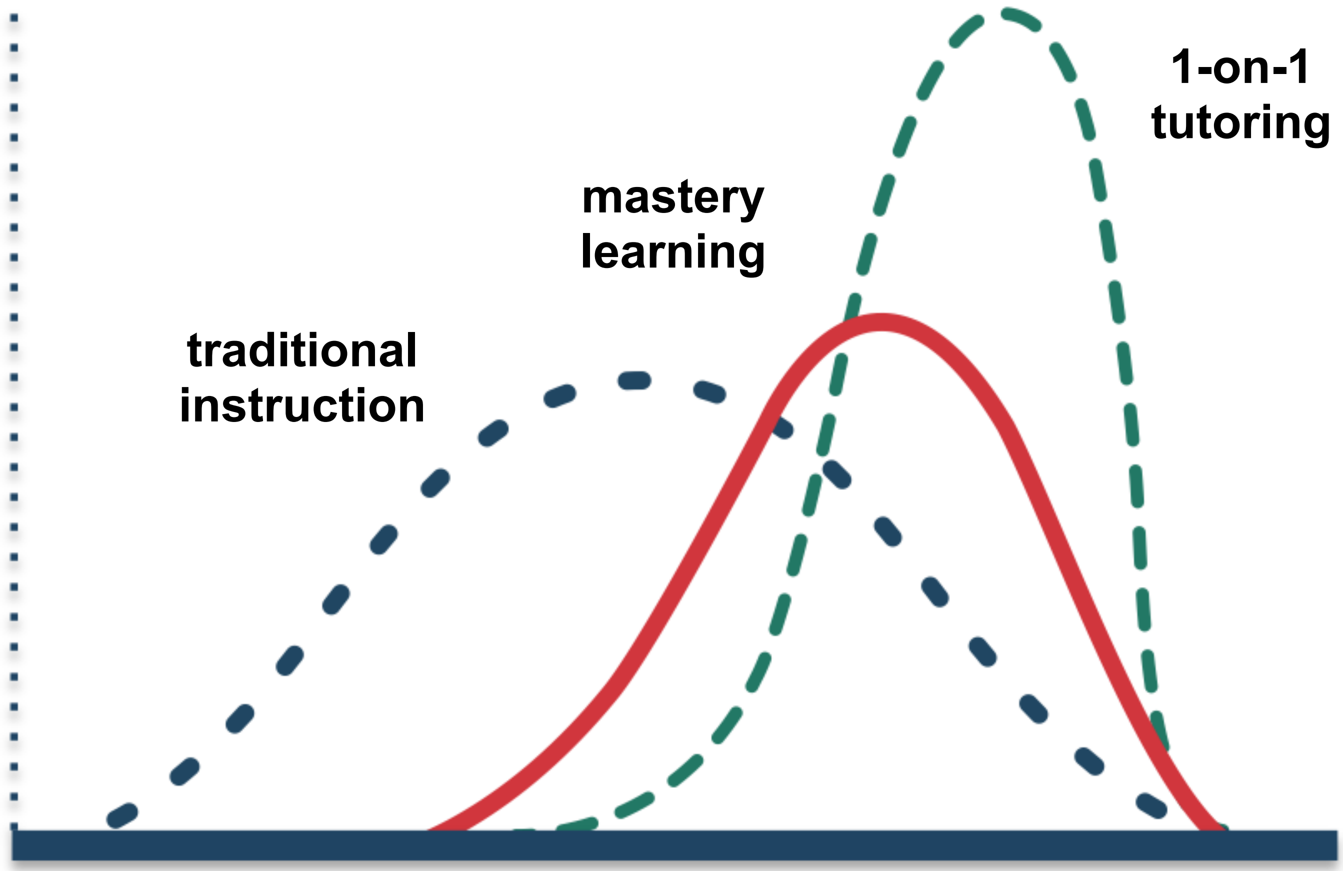
Self-driven mastery in MOOCs

9 July 2013



"The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring." B. Bloom, *Educational Researcher* (1984).

Elements of mastery learning

1. Predefined standards for mastery
2. Individualized, corrective feedback
3. Repeated formative assessments
4. *Gated progression

Elements of mastery learning

1. Predefined standards for mastery
2. Individualized, corrective feedback
3. Repeated formative assessments
4. *Gated progression

self-driven mastery

Goal of this talk

What can we say about **self-driven mastery** in massive open online courses (MOOCs)?

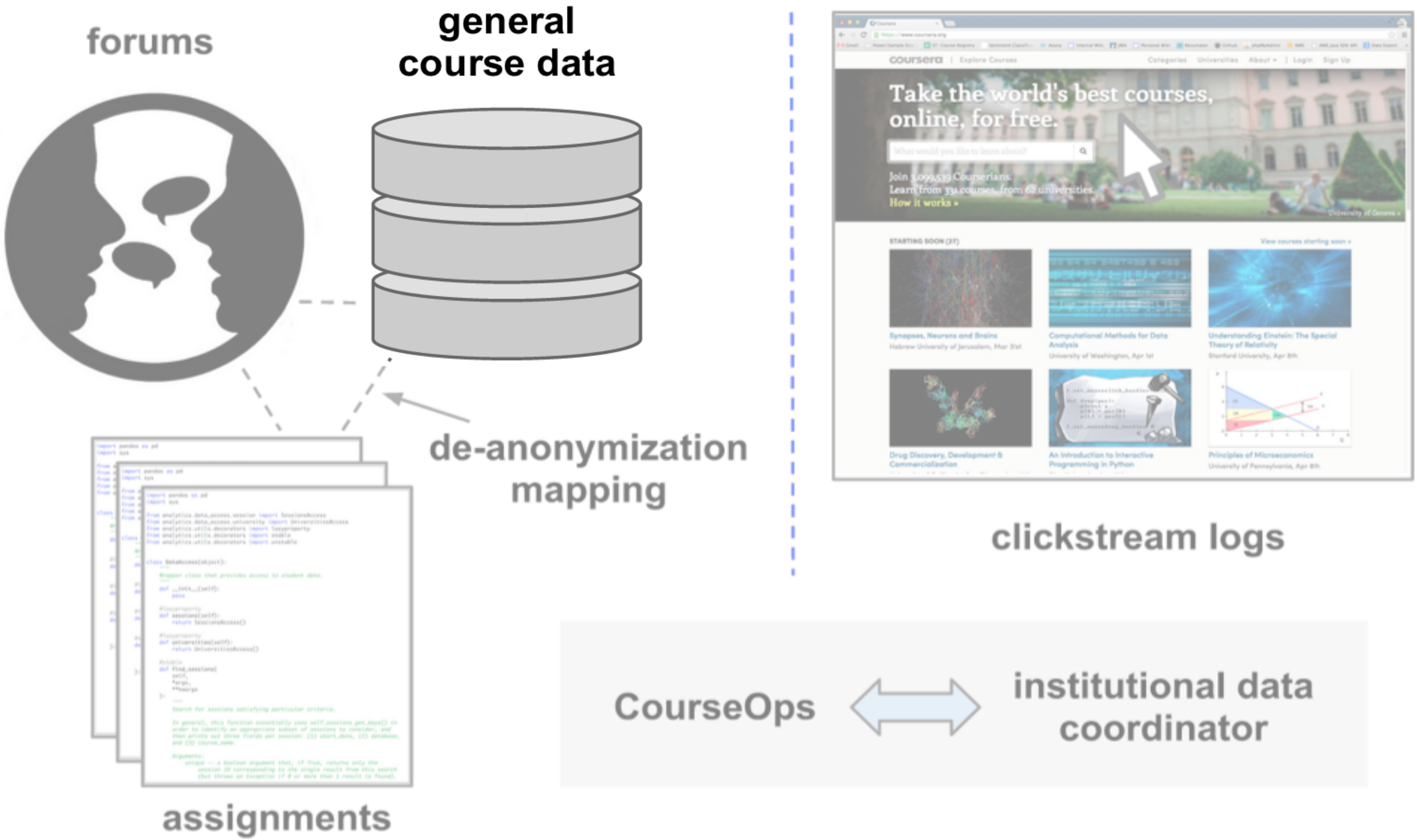
Course selection criteria

- formative assessments (e.g., quizzes and homework assignments)
- multiple allowed re-submission attempts
- summative assessment (e.g., final exam)

28 Coursera MOOCs

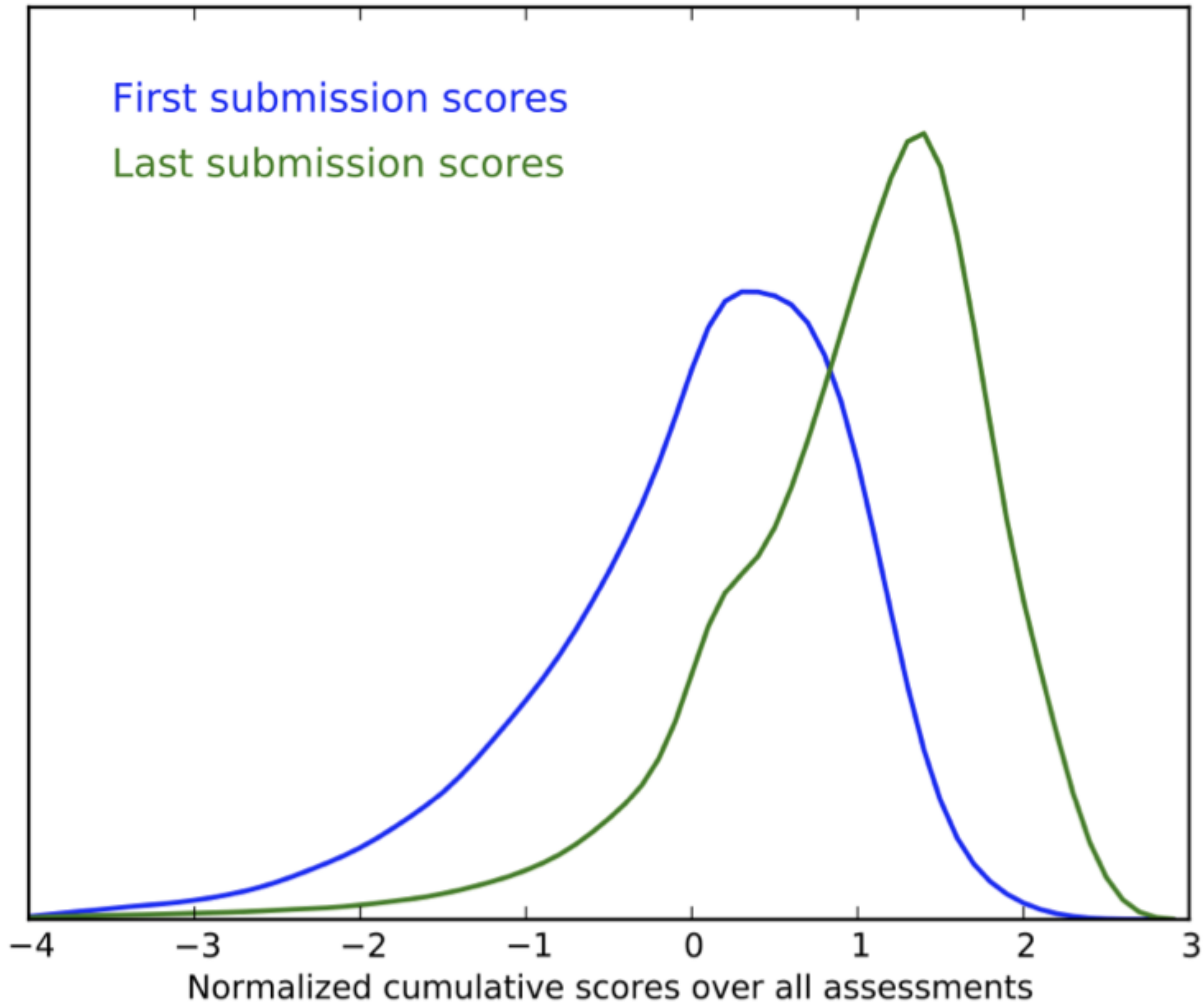
1 million students

3/4 STEM, 1/4 humanities

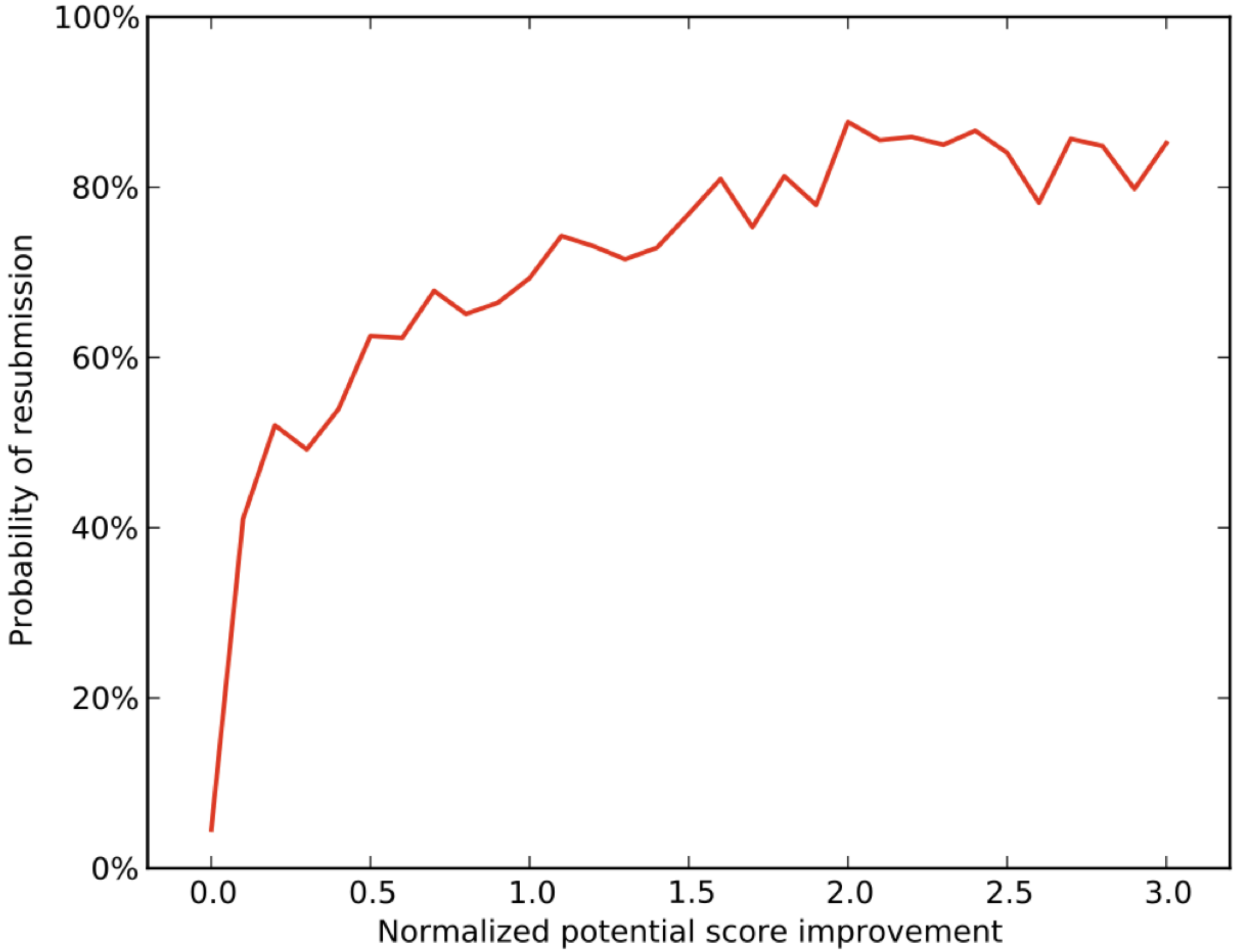


forums
general course data
de-anonymization mapping
assignments
clickstream logs
Take the world's best courses, online, for free.
How it works »
CourseOps
institutional data coordinator

Multiple submissions encourage grade improvement



Grade improvement provides incentive to resubmit



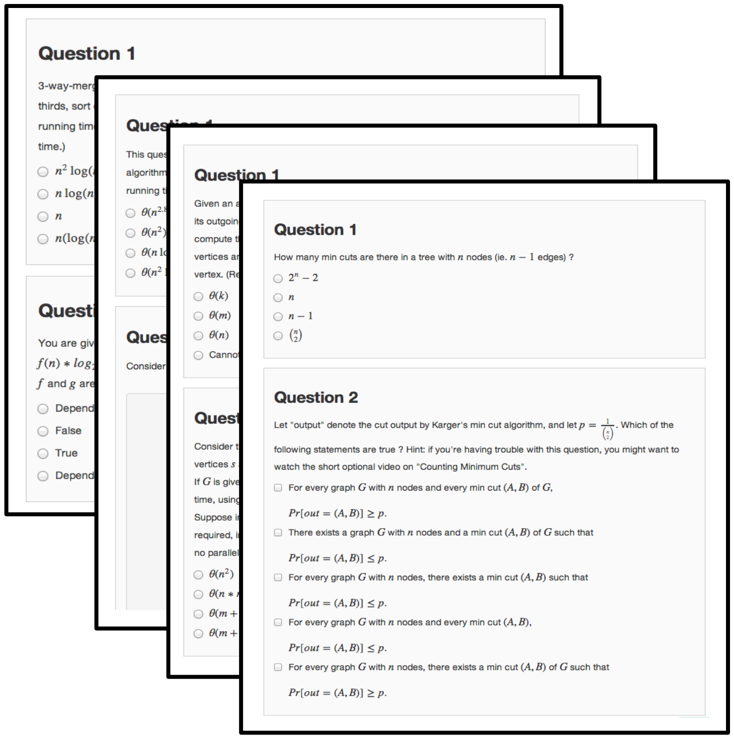
In some sense, this is all obvious.

- Who wouldn't want better grades?

- The law of diminishing returns

But is real learning taking place?

Measuring self-driven mastery



Question 1

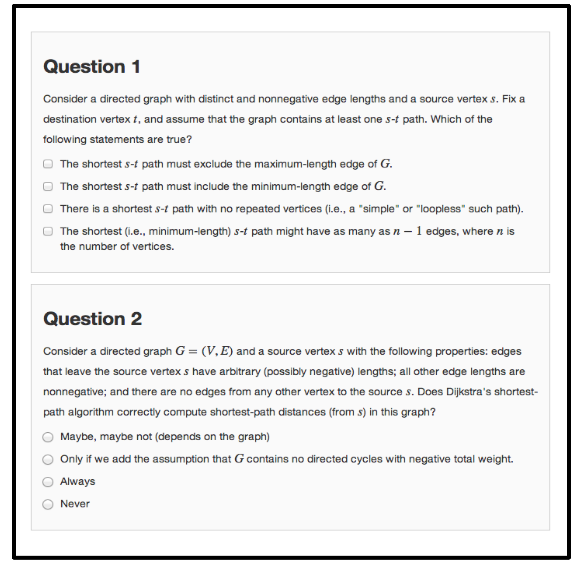
How many min cuts are there in a tree with n nodes (ie. $n-1$ edges) ?

- $2^n - 2$
- n
- $n-1$
- $\binom{n}{2}$

Question 2

Let "output" denote the cut output by Karger's min cut algorithm, and let $p = \frac{1}{\binom{n}{2}}$. Which of the following statements are true ? Hint: if you're having trouble with this question, you might want to watch the short optional video on "Counting Minimum Cuts".

- For every graph G with n nodes and every min cut (A, B) of G, $Pr[out = (A, B)] \geq p$.
- There exists a graph G with n nodes and a min cut (A, B) of G such that $Pr[out = (A, B)] \leq p$.
- For every graph G with n nodes, there exists a min cut (A, B) such that $Pr[out = (A, B)] \leq p$.
- For every graph G with n nodes and every min cut (A, B), $Pr[out = (A, B)] \leq p$.
- For every graph G with n nodes, there exists a min cut (A, B) of G such that $Pr[out = (A, B)] \geq p$.



Question 1

Consider a directed graph with distinct and nonnegative edge lengths and a source vertex s. Fix a destination vertex t, and assume that the graph contains at least one s-t path. Which of the following statements are true?

- The shortest s-t path must exclude the maximum-length edge of G.
- The shortest s-t path must include the minimum-length edge of G.
- There is a shortest s-t path with no repeated vertices (i.e., a "simple" or "loopless" such path).
- The shortest (i.e., minimum-length) s-t path might have as many as $n-1$ edges, where n is the number of vertices.

Question 2

Consider a directed graph $G = (V, E)$ and a source vertex s with the following properties: edges that leave the source vertex s have arbitrary (possibly negative) lengths; all other edge lengths are nonnegative; and there are no edges from any other vertex to the source s. Does Dijkstra's shortest-path algorithm correctly compute shortest-path distances (from s) in this graph?

- Maybe, maybe not (depends on the graph)
- Only if we add the assumption that G contains no directed cycles with negative total weight.
- Always
- Never

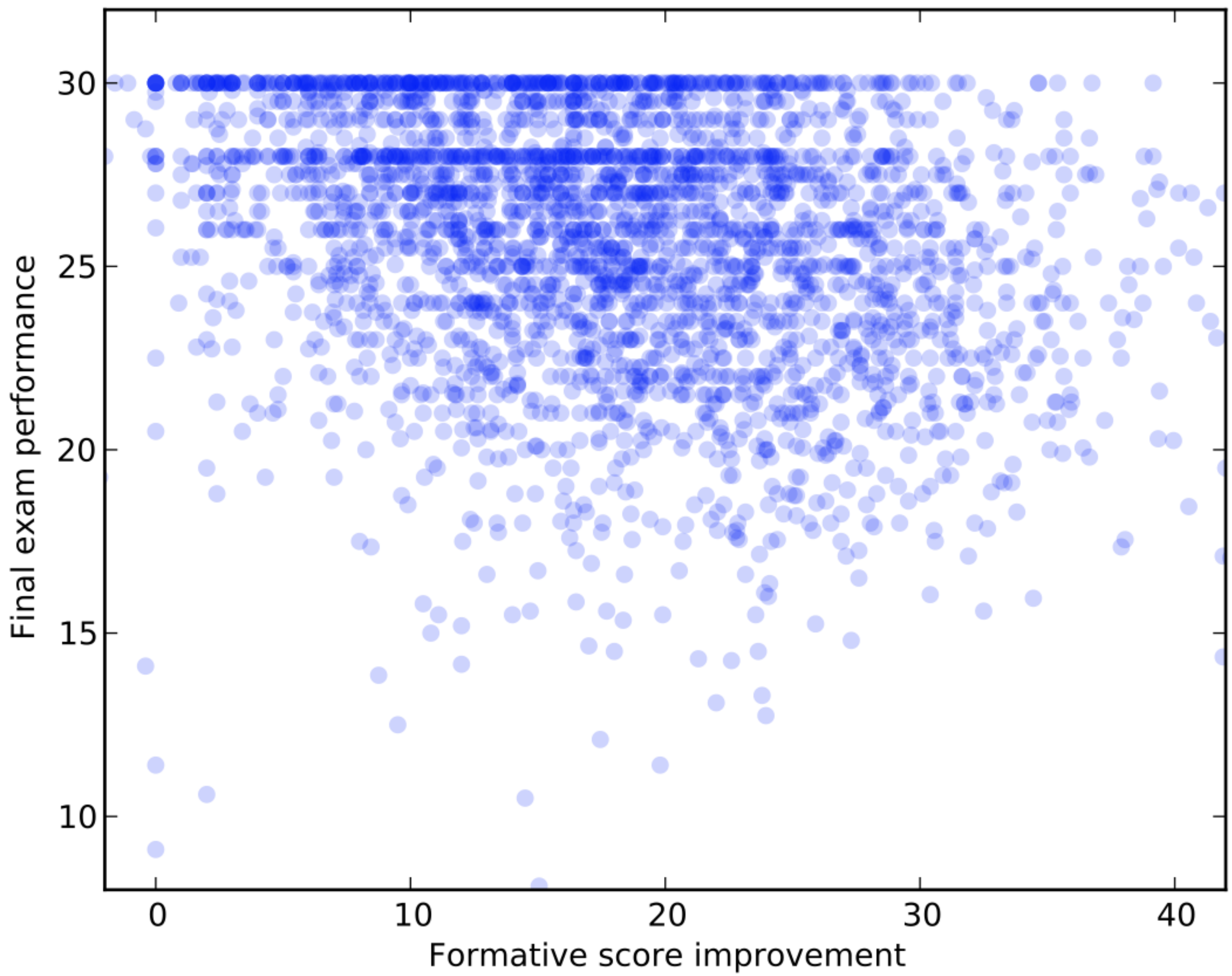
formative assessments

summed differences between first and last submission scores

summative assessment

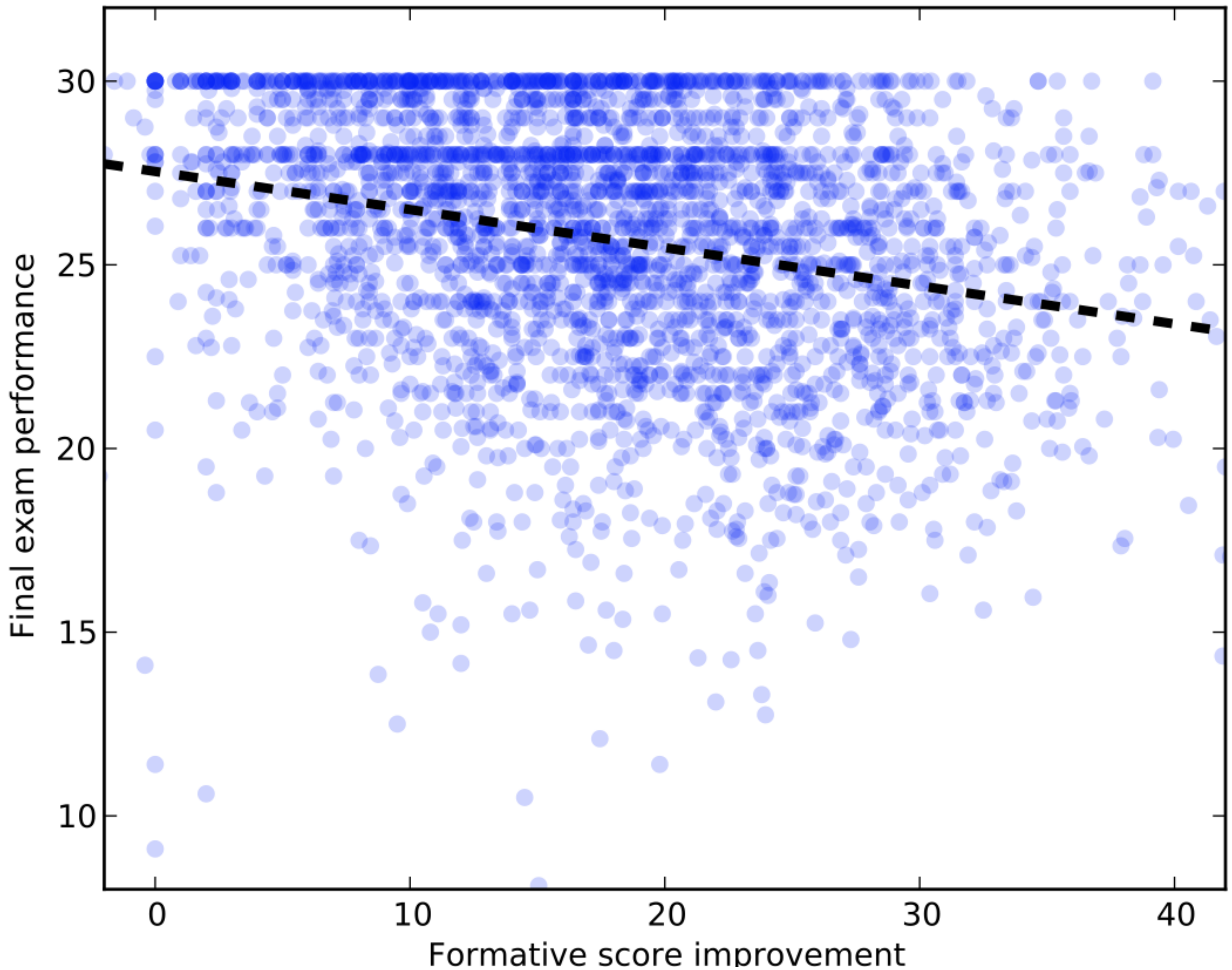
performance on final exam

Self-driven mastery



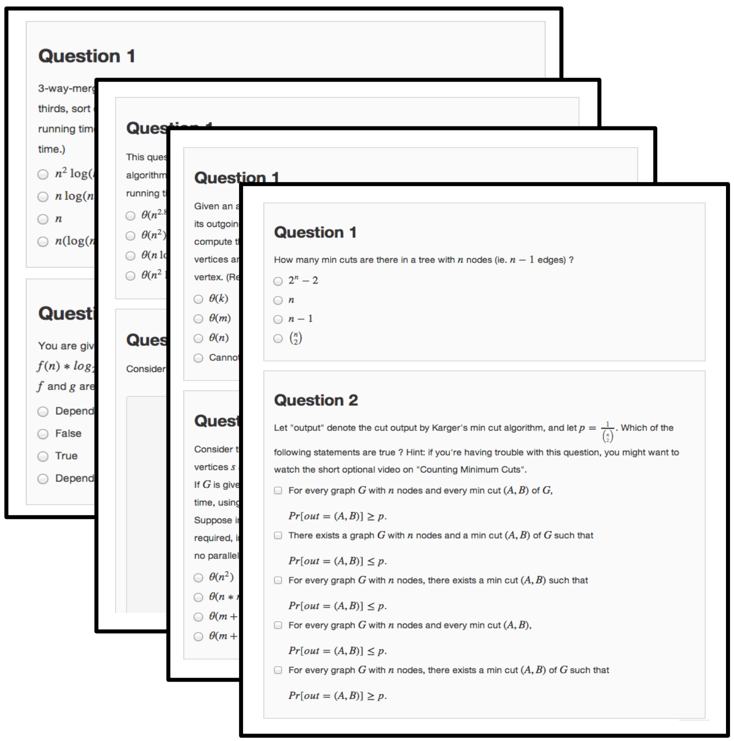
Algorithms: Design and Analysis, Part I
(Tim Roughgarden, Stanford)

Self-driven mastery



Algorithms: Design and Analysis, Part I
(Tim Roughgarden, Stanford)

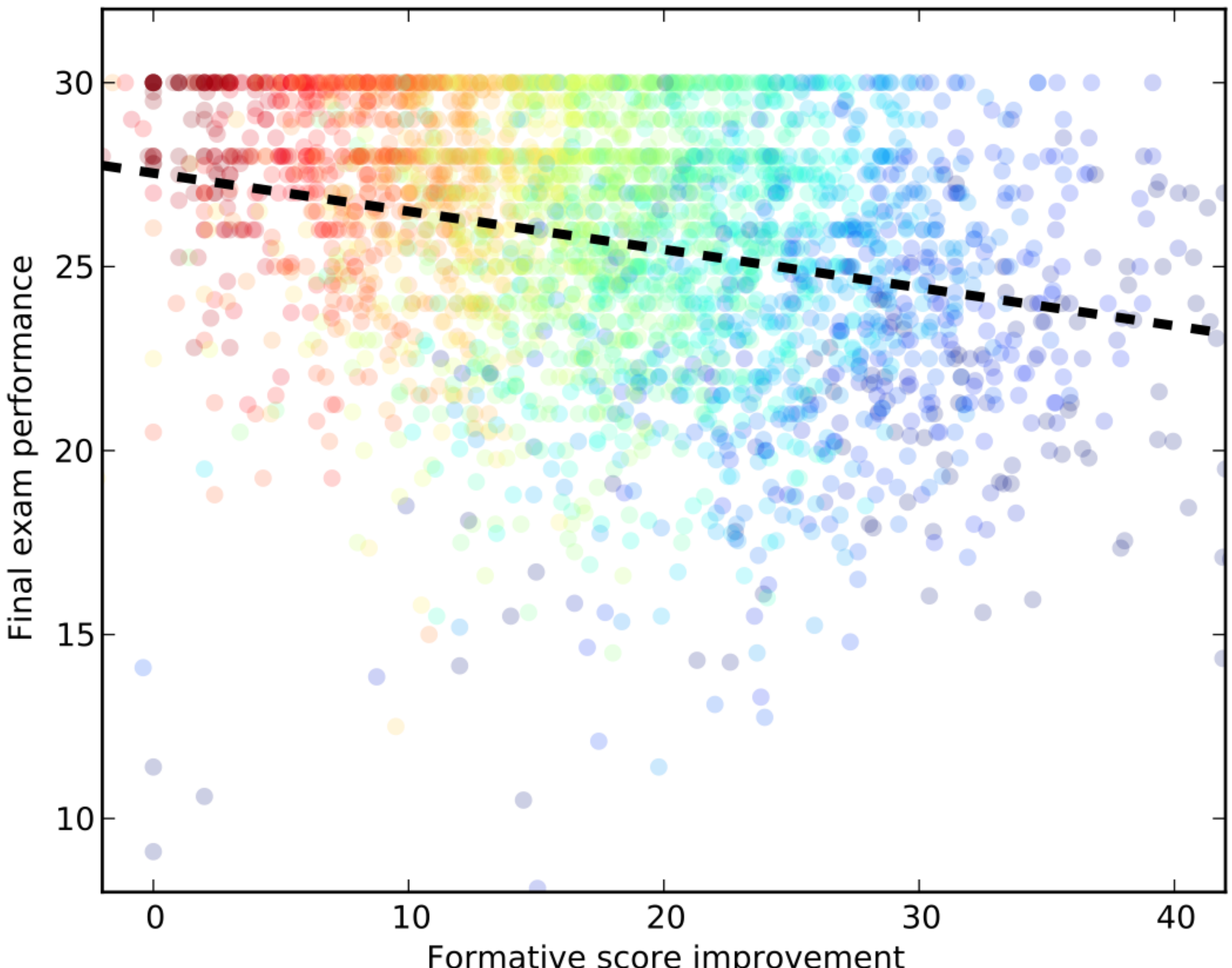
Covariate adjustment



<u>Include covariates to remove confounding by general ability</u>

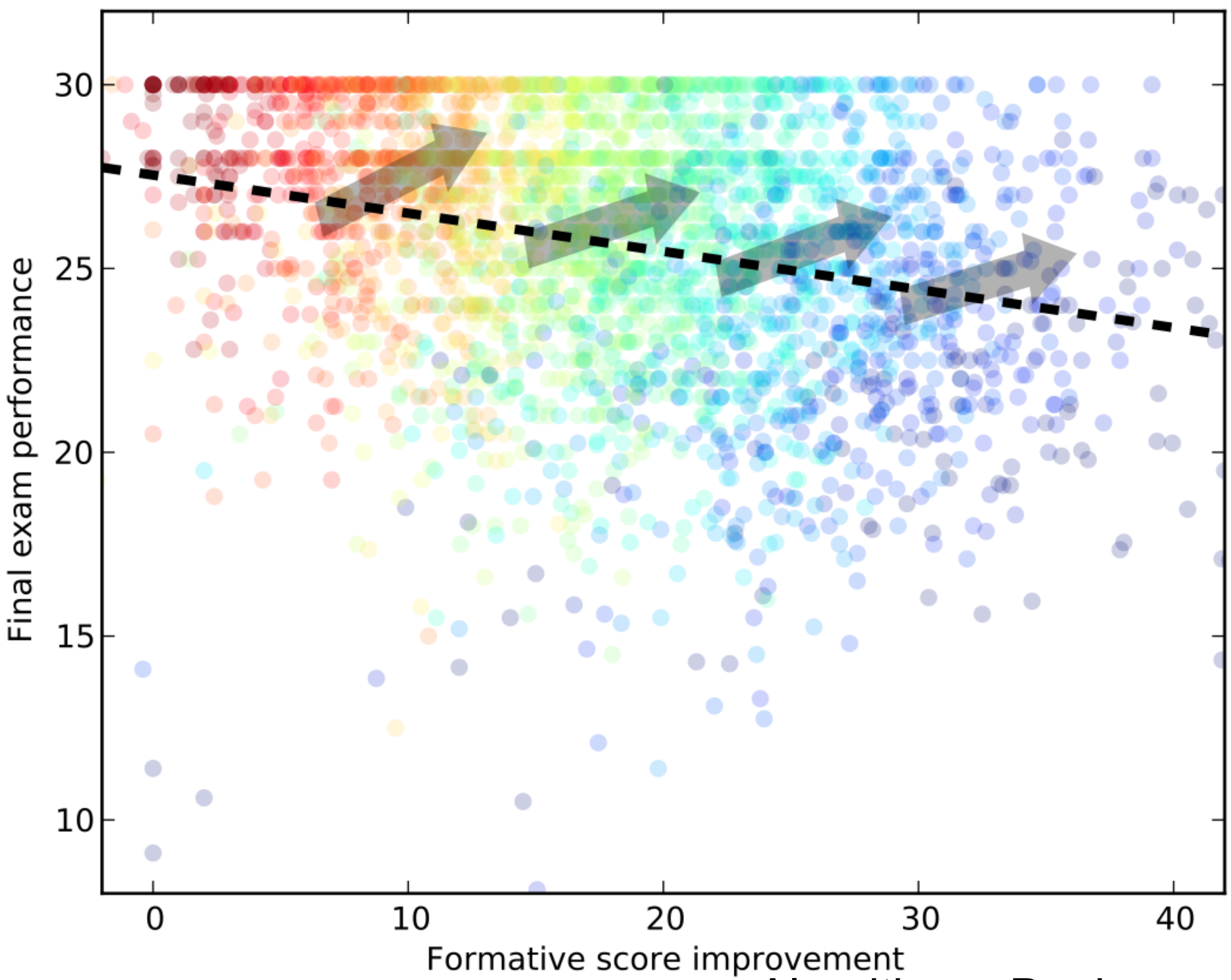
- summed initial scores for each assignment
- total number of submissions for each assignment

Stratified analysis



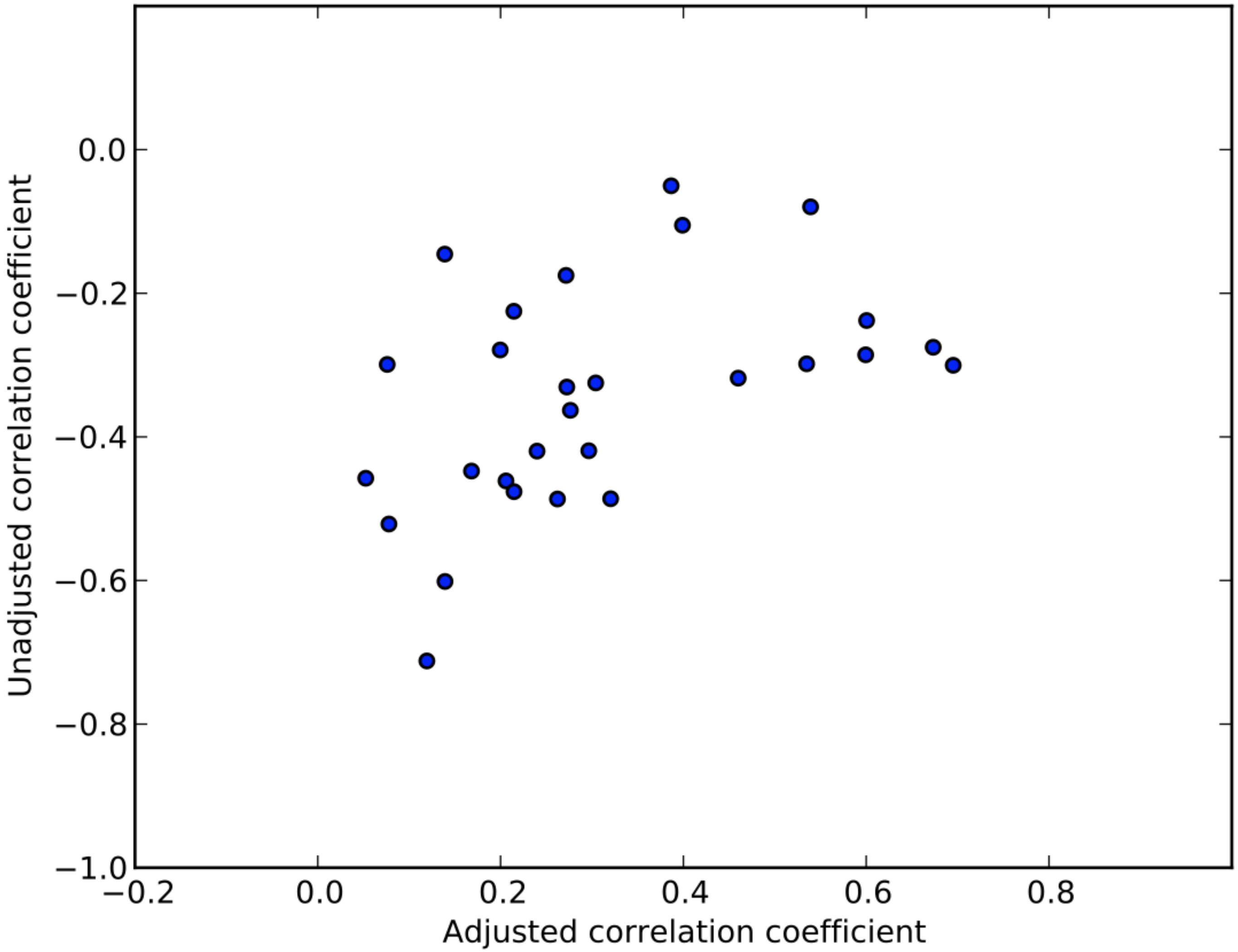
Algorithms: Design and Analysis, Part I
(Tim Roughgarden, Stanford)

Stratified analysis



Algorithms: Design and Analysis, Part I
(Tim Roughgarden, Stanford)

Stratified analysis



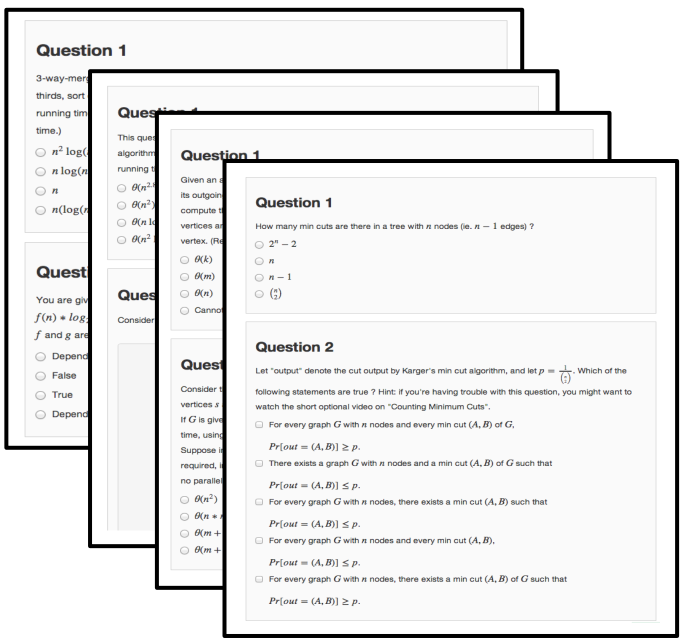
QED

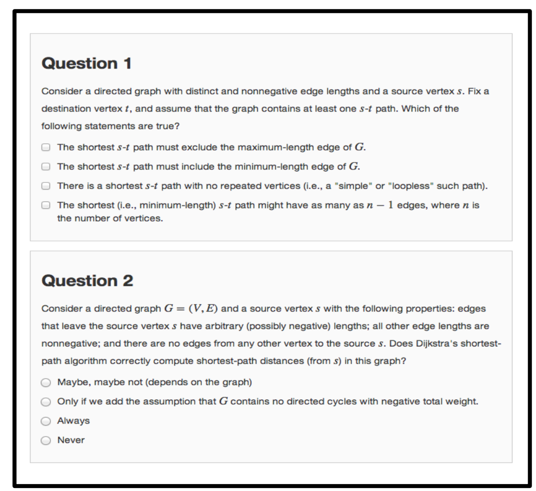
QED…or not.

Why we're being too optimistic

<u>Correlation vs. causation</u>

- Students who work the hardest to improve their grades are more motivated in general.





Question 1

Consider a directed graph with distinct and nonnegative edge lengths and a source vertex s. Fix a destination vertex t, and assume that the graph contains at least one s-t path. Which of the following statements are true?

- The shortest s-t path must exclude the maximum-length edge of G.
- The shortest s-t path must include the minimum-length edge of G.
- There is a shortest s-t path with no repeated vertices (i.e., a "simple" or "loopless" such path).
- The shortest (i.e., minimum-length) s-t path might have as many as $n-1$ edges, where n is the number of vertices.

Question 2

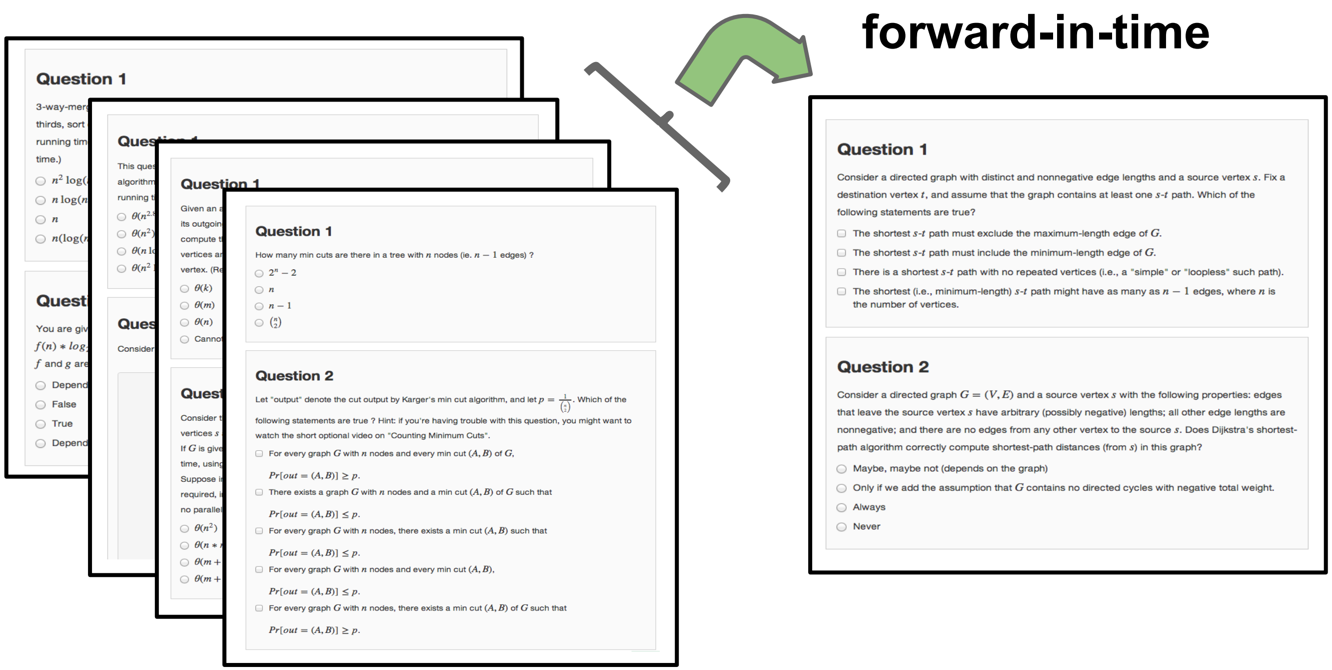
Consider a directed graph $G = (V, E)$ and a source vertex s with the following properties: edges that leave the source vertex s have arbitrary (possibly negative) lengths; all other edge lengths are nonnegative; and there are no edges from any other vertex to the source s. Does Dijkstra's shortest-path algorithm correctly compute shortest-path distances (from s) in this graph?

- Maybe, maybe not (depends on the graph)
- Only if we add the assumption that G contains no directed cycles with negative total weight.
- Always
- Never

Why we're being too optimistic

<u>Correlation vs. causation</u>

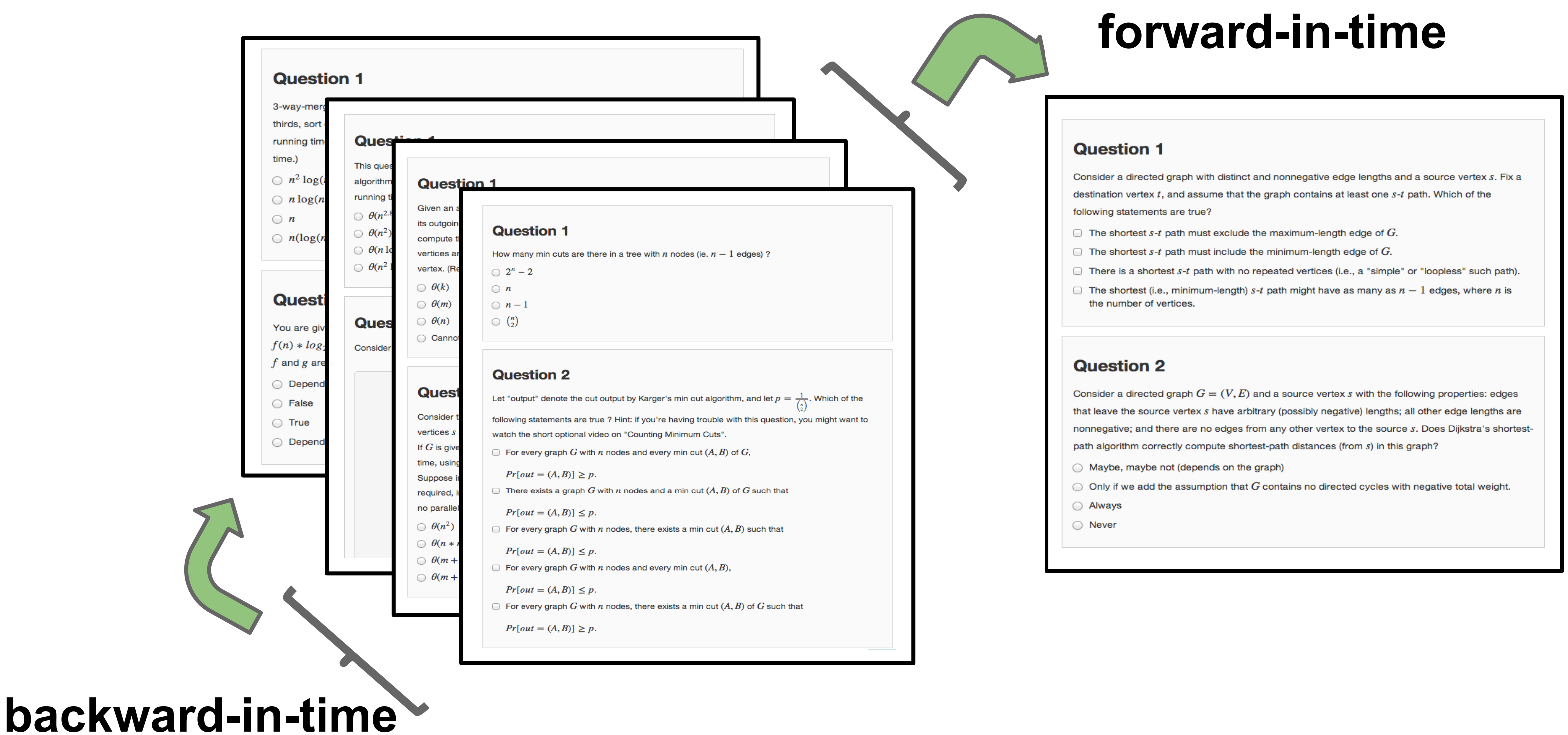
- Students who work the hardest to improve their grades are more motivated in general.



Why we're being too optimistic

<u>Correlation vs. causation</u>

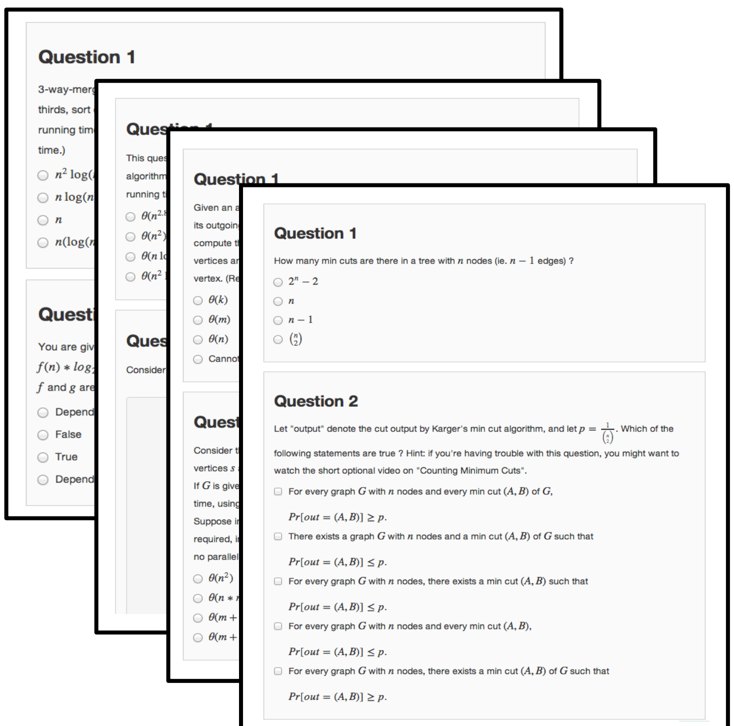
- Students who work the hardest to improve their grades are more motivated in general.



Why we're being too pessimistic

Ignoring cumulative effects

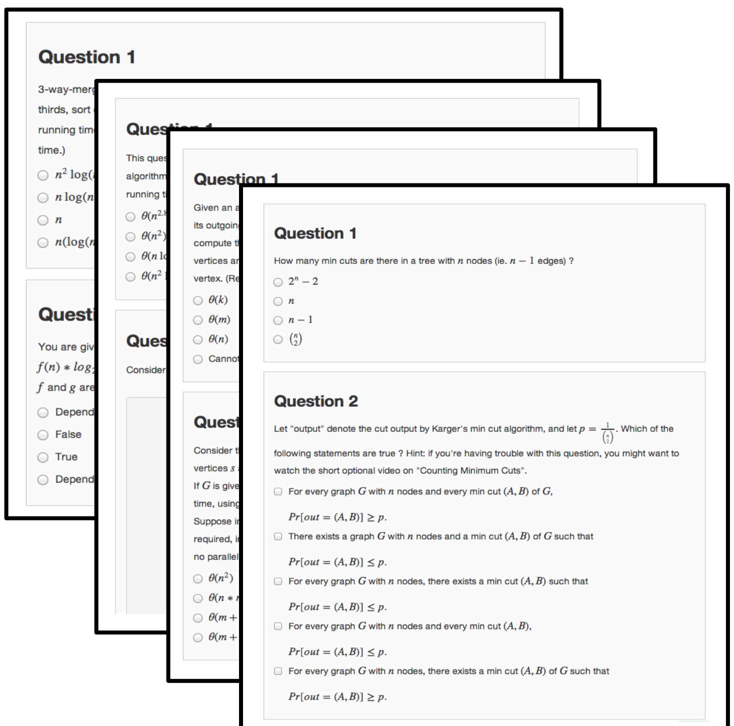
- By adjusting for initial score on each assignment, we won't detect cumulative benefits of self-driven mastery.



Why we're being too pessimistic

Ignoring cumulative effects

- By adjusting for initial score on each assignment, we won't detect cumulative benefits of self-driven mastery.



What if material from assignments build on each other?

- skills-based classes
- programming classes

Fundamental limitation

students who **do** make use of self-driven mastery

students who **don't** make use of self-driven mastery

Why not just implement mastery learning directly?

- Works in the classroom.
- Works in 1-on-1 tutoring.
- But does it work in the MOOC setting?
    - How do you know when mastery achieved?
    - How to set the threshold of mastery?
    - What if students get frustrated and quit?
    - What about students who don't want to play?

Summary

- Self-driven mastery
  - When adjusting for student ability, formative score improvements are associated with increased summative performance.
  - But beware the caveats!
- Mastery in the MOOC setting

Questions?

Chuong (Tom) Do
(cdo@coursera.org)