Teaching Reasoning and Proofs

. . . Even when they don't want to learn!

Outline

Target Audience

We teach courses of all kinds: from required freshman course (251) to specialized elective for upperclassmen (machine learning).

- Big difference between Freshmen and Seniors
- Big difference between 251 and electives

My points will be most relevant to 151,251-like courses.

But they often extend to other courses, other paradigms.

Disclaimer

Many of the things we'll be discussing have no correct answer.

And they often don't generalize well.

My goal here is to provide you with the insights I've gained over the past several years. If I suggest something in this talk, it means I have at least anecdotal evidence that it works.

If you take one thing away from this talk, it should be that the more active you can make your students in the learning process, the better they will learn. Countless studies indicate this across various domains.

I believe in this so much that I can't get through a talk without involving the audience.

Let's consider some "student answers". Some of you will have seen some of these before.

Question: What is $|\{☺, ☺\}|$?
Answer: 4.

Question: Prove that the irrational numbers are uncountable.
Answer: We know the irrational numbers are exactly $\mathbb{R} \setminus \mathbb{Z}$; so, . . .

Question: Find the probability that. . .
Answer: . . . so, the probability is $\frac{4 \cdot 5}{10}$.

Question: Is $(\mathbb{Z}, -)$ a group?
Answer (on quiz): Yes.
Answer (on exam): Yes.
Answer (on final): Yes.

Every Action...

Your Action: In recitation, you tell students "remember, don't prove a statement by starting from what you want to prove and going backwards. It's considered bad design/style.
Student Result: More people do this than before!

Your Action: Work hard to give good recitation. Do your best.
Student Result: Your recitation average is among the worst.

Your Action: Tell them what to study for the exam explicitly.
Student Result: After getting their exams back, they tell you that the exam was on things they weren't expecting.

Your role as a TA for a theory/algorithms class varies drastically per student. You could be. . .

- A sounding board **for the student who is really struggling**.
- A source of interesting material/problems **for the student who wants to go further**.
- A kick in the ass **for the student who is more concerned about their grade than learning**.
- A wake-up call **for the CS freshman who thinks he's smarter than everyone else in the building**.
- etc.

Giving the types of recitations we tend to give ("here's a bunch of problems, let's solve them!"), there are some major differences (and unique challenges!) that arise:

- They actually show up!
- It's hard to assess who is actually solving things.
- Managing many groups of students trying to solve problems can be intimidating.
- You don't know the answer to their question right away.
- Their skill is very unbalanced (more so than normal)... to the point where every problem is wrong for about half the class.

And then there's problems that often show up in any recitation:

- They won't say anything... okay, except for that one kid!
- Dealing with a change in the trajectory of the recitation.
- I screwed up... what now...

- Can you read my proof?
- What's the answer?
- Do I have to justify x?
- I don't think I'm going to finish the question in time, can I have an extension?
- I need a hint.

When you're asked one of these questions, the goal is to reinvent the question into one that you can answer.

Outline