Teaching Statement
Prateek Bhakta

I've enjoyed teaching for as long as I can remember. My early practice tutoring in middle school and high school taught me both patience and a love for teaching, and I developed an intuition for presenting complex ideas clearly. While I was a student at U.C. Berkeley, I learned most of my skills as a teacher while working at a university run tutoring service that provided free homework help and organized recitation-style study groups outside of class. There, my boss and mentor taught me how to adapt to different learning styles, quickly identify where students are stuck and what they are missing, and teach them to make key insights for themselves with as little coaching as possible. I attended professors' lectures for the courses I supported, gave my own lectures to stress important concepts and fill in gaps, and created supplemental problem sets to give students extra practice. From my unusual viewpoint, I was able to learn from the strengths and weaknesses of many different professors' teaching habits. I grew very attached to my tutoring work, and still found time to tutor there in the mornings while working full time at a nearby start up.

After entering grad school at Georgia Tech, I quickly found that I missed teaching, and I looked for every opportunity to teach that I could. I taught Python to interested grad students, gave tutorials on the basic tools of my research, and served as a teaching assistant several more times than required. Because of my positive evaluations, I was asked to teach the introductory algorithms course as head instructor this coming Spring, which is very rare for PhD students in our department.

Through my experience, I've identified three important goals that are critical when teaching the creative problem solving concepts essential for math and computer science.

Course Design and Content Mastery. In most advanced math and computer science courses, it is more important to develop intuition for strategic problem solving than it is to practice the mechanics of carrying out a calculation. Many students will inevitably forget specific solutions to various problems, but my goal is for the high level strategies and structure to remain with them long after they graduate. This intuition is difficult to teach solely through lecture, and the most effective way to learn this is with guided practice in homework, both alone and with other students.

At Georgia Tech, I was a TA to a senior graduate student who was teaching her first course, and we used my teaching experience to plan the curriculum together. I wrote assignments to develop my students' critical thinking skills by giving them carefully scaffolded questions with steadily more difficult problems. One effective strategy that I used is to require groundwork-building homework questions to be done individually, while allowing collaboration on the more conceptually difficult questions. This encourages all students to contribute when working together, and it avoids situations where the work is done too unequally. I also introduced fun and challenging bonus questions on homeworks and exams to encourage students to think in new and complex ways, and combine concepts from multiple parts of the course. These bonus questions helped keep more advanced students interested in the course, without alienating the students who were struggling.

This balance was especially challenging to achieve when I was teaching middle school students at a summer camp in Korea. Due to some miscommunication, I was instructing children as young as eight years old as well as the more advanced middle and high school students that I prepared for. I had to lecture and design a curriculum on the fly that would suit all my students in some fashion, all while contending with a difficult language barrier. I patiently taught advanced ideas in simple terms, so that the younger students were at least exposed to new problem solving methods. One of my most memorable successes as a teacher occurred when, after carefully explaining the notion of countability, I asked if the set of all pairs of integers is countable. This is not usually introduced until sophomore college math courses, but eventually a nine-year old student who barely spoke English figured it out by himself, and he showed the class how it worked with a simple drawing!

Awareness of Student Background and Understanding. I always strive to keep mindful of what my students understand and what they will be capable of learning given their current understanding. Too often, professors lecture blindly, teaching concepts that they deem pedagogically important without paying attention to the students themselves, especially at larger universities. I witnessed one memorable example when I was supporting a probability course. The professor spent a day lecturing about the philosophical meaning of probability and chance in mathematics, incorrectly believing that this was the reason for his students' poor understanding. I knew from my tutoring sessions that the students were just unfamiliar with the math tricks involved in the computations. He not only wasted a day of lecture, but confused his students all the more. It seems as if many professors make this mistake at some level; they teach as if instructing a younger version of *themselves*, focusing on the particular roadblocks that they themselves had. This approach neglects the myriad other issues that people could have when learning, and fails to account for the fact that the professor's learning experience was almost certainly exceptional.

My many years tutoring and grading "in the trenches" give me both a familiarity with the actual roadblocks that students regularly struggle with and experience presenting material within the right frameworks to help surpass them. As a teaching assistant, I strove to identify what students were not grasping in their lectures by analyzing their homework responses, their difficulties in office hours, and their questions on an online discussion forum that I set up. I then worked with the instructor to better adapt the lectures and homeworks to the students' needs.

Attention and Motivation Both in lecture and when studying, I've often seen that when someone believes that they will not be able to understand the material before them in the near future, then they will likely "zone out" and stop thinking and learning for the near future. This is often triggered by tedious complexity, losing the flow of the lecture, or getting stuck trying to understand a previous idea. It is made worse when students don't understand *why* the material that they are learning is important, or how it fits into a larger picture. Students who are discouraged for too long will often give up, or worse will occasionally cheat. This is especially important problem when learning creative problem solving, as focusing on the entire scope of an analysis is often essential to understanding how and more importantly why our problem solving tools are applied.

In the past, I've used many approaches to help students overcome this common problem. I take at least one short break in the middle of class periods to allow students to gather their thoughts, and this tend to help them follow the rest of the class much better. I make sure to motivate the problem solving tools that I teach by giving examples of real world applications and explaining how these tools will fit into the larger goals of the course. In addition to office hours, I've found that providing an online discussion group helps students who would otherwise be reluctant to seek help. Finally, I try to keep a lively and interested tone when lecturing, and I engage my students during lecture as much as possible by encouraging them to answer questions and fill in details.

When I see that students are struggling, I make sure to acknowledge that the material I am presenting can be difficult to mitigate discouragement and help them back on the path to understanding. This builds the confidence of students new to computer science who might otherwise feel that they might never catch up to their more experienced peers. This is essential to promoting diversity in math and computer science, and it improves retention in STEM majors in general.

I sincerely enjoy my role as a teacher, and want to have the same impact on my students that many of my teachers have had on me. Besides the more common computer science classes that I have experience with, I look forward to designing more specialized courses in fields like complexity theory and randomized algorithms. I hope to keep students engaged and interested by accommodating students at different levels of experience, and I will strive to develop their problem solving skills inside the classroom so that they will be effective critical thinkers beyond the classroom.