Teaching Statement

Kaiyu Yang

I believe effective teaching in computer science should encompass four goals: (1) help students build a firm understanding of the fundamental concepts; (2) foster critical thinking and problem-solving skills that allow students to extrapolate beyond the basics to any new challenges; (3) help students grasp the big picture of the field and prepare them for critical choices they have to make about studies and careers; and (4) cultivate their interests and intrinsic passion about computer science, which will help them thrive and be creative in the long term.

Fundamental concepts are important, but learning them is laborious and can be painful without proper guidance. One technique that I find useful is active learning—students understand materials better when engaging with them actively, e.g., through discussions. When walking through problem sets, instead of simply presenting the solution, I ask students to work on them and explain their solutions to the class. I also make frequent use of 5-minute discussions in small groups and have students summarize the discussion. Feedback from students indicates these strategies help them learn more effectively. When presenting the materials, one challenge is accommodating students of diverse backgrounds and learning styles. I’ll elaborate more on my approach in the diversity statement.

I strive to prepare students for novel challenges that they will encounter in the future. These challenges may go well beyond the basic principles taught in classes but require critical thinking and problem-solving skills. These skills can be trained in classes by guiding students through the thinking process of recognizing and formulating a problem, then applying existing knowledge in novel ways to derive the solution. For example, when designing problem sets, I avoid trivial problems testing memorization and include problems focusing on creative applications of knowledge. Another strategy is hands-on projects. While I was a teaching assistant in the natural language processing class, I designed programming projects for students. And I was amazed by the independent problem-solving skills the students demonstrated without detailed step-by-step instructions.

“How can I find an internship for the winter break?” asked my niece, a computer science major who has just finished her first semester in college. When I asked why she wanted an internship and what she had learned about computer science, she replied, “Hmm..., I’m not sure, but I have learned some Java.” This episode made me reflect on the importance of big pictures as roadmaps for students to make informed decisions about their studies and careers. In my teaching, I’d like to help students develop this big picture by connecting the materials with what they have learned and will have to learn in other courses. If students can put the materials in a clear context, it helps them not only see the big picture but also learn the materials more effectively. Further, during office hours, I’m happy to chat more broadly about any topic in computer science that I may not cover in classes.

Ten years ago, I entered college as an electrical engineering major but switched to computer science after a few coding projects, because they were fun. And I still believe, if there is one thing for thriving in computer science, it is passion. First-class computer science education sparks and cultivates such passion. In my teaching, I strive to offer the maximum flexibility for the students to do what they want within the course’s requirements. In the natural language processing class, the final project was on reproducing published papers; however, there was one group of students who preferred to pursue their own research ideas instead. I encouraged them to do so; and paid particular attention to monitoring their progress to ensure they were on the right track.

As researchers, we teach not only in classrooms but also through research mentoring. During my Ph.D., I have mentored multiple junior students, including Jacqueline Yau—a Master’s student from Stanford. Our mentoring relationship was also a learning process for me. The biggest lesson was to reconcile two mostly aligned but occasionally competing goals: (1) making progress on the project and (2) training junior researchers. I believe the outcome of a project includes not only publications but also trained junior researchers—which may be even more valuable in the long term. Therefore, whenever possible, I engage students in intellectual aspects of our work beyond simple coding implementation, even if that does not contribute to the short-term progress. Luckily, we had success in achieving both goals. Our collaboration led to a submission to a top-tier conference. And after walking through conceptualizing research ideas, implementing them, and writing research papers, Jacqueline decided that she wanted to pursue research further in Ph.D. programs.