Teaching Statement
Qizhen Zhang

When I started my undergraduate study, I knew little about computer science. Through the learning process, my teachers provided tremendous guidance and help. In my third year, the professor in my distributed systems course introduced me to systems research and that was where all my excitement about research started. Later, I decided to make computer science my life career. Spreading the knowledge I have acquired and inspiring future computer scientists are top motivations for me to stay in academia, and I have appreciated the opportunities in graduate school for me to teach and mentor junior students.

Massive online teaching. In 2019, our school launched the online Master of Computer and Information Technology (MCIT) program, a fully-online master’s degree program in Computer Science. I participated in the program as a course designer and teaching assistant (TA) for the Computer Systems Programming course, which covers fundamental concepts in computer systems including operating systems and networked systems. The program targets at students who have little engineering background, so it adds extra requirements on controlling course difficulty and guaranteeing content clarity for both teachers and TAs.

As a course designer, I was in charge of designing and implementing the third and most challenging programming assignment: network programming with sockets. This assignment requires students to apply various concepts they have learned in the course such as memory management, threads, event-driven techniques, and networking. I decided to divide the assignment into three parts for testing basic socket programming, multi-threading, and event-driven programming respectively. I wrote the auto-graders for all three parts and improved their robustness for handling various corner cases hit by students’ submissions. To ease other TAs’ work, I recorded a video to demonstrate the implementation of the auto-graders and how to debug students’ submissions. The auto-graders have served thousands of students and are still in use today.

As a TA, I held online office hours to answer student’s questions. To explain abstract and advanced concepts, I started from things that were concrete and basic and built explanations on what students knew. I also used live demonstrations as a necessary tool for students to fully understand what I taught.

On-campus teaching. I also served as a TA twice for Networked Systems, an on-campus graduate-level course. Rather than focus on a specific teaching task, I actively took part in all stages: attending the classes, holding office hours, and grading projects, mid-terms, and final exams. This course covers all layers in the classic networking stack, from physical links to transport protocols and applications, as well as recent network innovations. In particular, there was a series of projects that used P4, a domain-specific language for programmable switches. Students were struggling as P4 was unlike any of the languages they had previously interacted with. In my office hours, I explained to students the overall switch architecture and how a packet is exactly processed on a switch. As soon as students knew why the language is designed in its way, it became much easier for them to proceed. I really enjoyed such interactions as I had chances to listen to students’ perspectives and improve my answers based on their feedback.

Teaching philosophy. I view all human knowledge as a massive graph. Researchers strive to expand this graph by creating new dots and connecting them to existing ones, and teachers endeavor to let their audiences understand and utilize some parts of it. Hence, I see teaching as a complementary effort to research that implements the actual impacts of knowledge. Like researchers create new dots based on current ones, teachers should teach dots that can directly link to the ones students have grasped. Students can be fast or slow at understanding specific dots, but better contexts and material connections can optimize the length and excitingness of the overall learning process. Teachers should also make sure that once students understand all the dots that are taught, they will have a good understanding of the whole subject with both depth and breadth. Thus, I view my task as a teacher as deciding how best to break the subject into these dots and order them properly.

Teaching interests. With the expertise and visions developed in my research and industrial experience, I am looking forward to teaching students the internals of database systems, big data systems, distributed systems, and computer networking, as well as the practices of building these systems and their advanced topics. In general, I am able to teach any course on data management, computer systems, and computer science foundations.

Mentoring. During my graduate study, I have closely worked with and mentored fifteen students for research: seven junior graduate students, seven undergraduates, and a high school student. I believe the most important factors in achieving success are passion and perseverance. Hence, my philosophy for mentoring is to empower
students based on their passions and enlighten them about the importance of perseverance when necessary. Different students are passionate about different topics. For example, some of the students I mentored like building efficient middleware big data systems, while others are more interested in low-level infrastructures such as operating system kernels. My approach is to provide students with as many options as possible, and let them select the topics that truly attract them. Perseverance is especially important for systems research, as the process of developing a novel system is often long and there can be many difficult moments. Helping students recognize the meanings of what they do and reminding them of the big pictures are as necessary as helping with technical details. The collaborations with the students I mentored so far have resulted in a number of publications in top-tier database and systems conferences like SIGMOD, VLDB, CIDR, CIKM, SIGCOMM, and SoCC. All undergraduates who have graduated are now Ph.D. students at Stanford, Princeton, Cornell, Peking University, as well as Penn.