I was very lucky to be given the opportunity to teach a course at Stanford in the summer of 2014. Branching out from my research on databases, I chose to teach CS161, Design and Analysis of Algorithms. The class had 69 students and I coordinated a team of three TAs. I was responsible for delivering two 110-minute lectures per week for eight weeks, preparing problem sets, a final exam, a programming project, and holding office hours. Before the course began, I reflected on the classes that I appreciated the most as a student, and identified three commonalities across them, which I did my best to include in CS161.

- **Assignments were hard and fun.** Homework problems need to be hard enough to push understanding deeper, but not so hard that students get completely stuck. I tried to design problems in multiple parts of increasing difficulty, with the intention that the last parts would be solvable only by the best students. To engage students who do not enjoy theory, I designed an intense two-week programming project, including a competition: the three student winners with the fastest implementations of an algorithmic challenge explained their programming tricks to the class.

- **The content was relevant beyond the academic setting.** CS161 covers standard algorithms and algorithmic analysis techniques that any computer scientist should know, so it did not require any effort to make the material applicable for students’ later careers. However, to make the material more immediately relevant, each problem set included a tricky algorithmic question asked by software companies in interviews. In my evaluations, many students said they appreciated those problems the most.

- **The lectures included stories and history.** I believe narrating personal stories and history behind technical material is a great way to inspire students and evoke enthusiasm. For example, students might find Strassen’s matrix multiplication algorithm an interesting application of the divide-and-conquer paradigm. But their interest will be piqued even more when they learn that it was the first subcubic matrix multiplication algorithm, Strassen mysteriously never revealed how he came up with it, and the current best-known matrix multiplication algorithm is due to Virginia Williams, who teaches at Stanford so the student can say hello to her in the hallway. For each algorithm I taught, I narrated its history and/or a fun story about the developer of the algorithm.

One challenging aspect of teaching CS161 was that it left almost no time for research. The intensity was due in large part to the fact that it was my first time teaching, although summer quarters at Stanford are more compact. As a faculty member, I will need to better balance my research and teaching.

In terms of teaching breadth, I am qualified to teach courses in database systems, database theory, data structures, introductory algorithms, randomized algorithms, and courses or seminars on large-scale distributed data-processing, as well as introductory computer science courses. I am especially interested in developing a project-based course in which students learn, apply, and contribute to the emerging technologies in today’s open-source distributed data-processing platforms.