Personal Statement

Jessica Chang

I am in graduate school because I enjoy studying the design and analysis of algorithms. I want to contribute to research in this area, and I want to help others understand the significance and appeal of this field.

Math has always been one of my favorite subjects; as early as elementary school, I appreciated the immediate objectivity of math above the subjectivity that seemed to exist in other academic fields. My introduction to computer science was fostered in a gender-balanced environment. By the time I experienced the reality of severely lop-sided gender ratios in computer science, it was too late; I was completely hooked. I enjoyed those courses so much that one thing led to another and I soon found myself studying computer science (and math) in college and eventually the design of algorithms.

A key event that influenced the path of my academic journey actually did not occur in academia; I spent a summer at Microsoft developing software and learning about the product life cycle. Most importantly, the internship provided a stark contrast between algorithmic problems that I had previously encountered and the problems faced in the software industry.

The following year, I took a course on the design of algorithms, which only confirmed the conclusions that I had reached earlier. My growing interest in the subject led to undergraduate research with Professor Samir Khuller.

Unlike the rigid structure of coursework, I have found that research allows for more creativity. It also demands stronger motivation, since research problems are not studied with the assurance that conventional methods are sufficient or even that a solution exists. I often started trying to solve one problem and ended up solving another; I set out to prove something for which I later found a counterexample. I have found research to be more involving and more rewarding than coursework. In general, taking a course cannot compare with amount that you learn through reading the results of a related paper or understanding a technique that may (or may not) prove directly useful in your research.

The impact that research in theoretical computer science has on society, of which scheduling is but an example, is profound. Scheduling problems arise virtually in any situation where vast amounts of information must be disseminated to clients in a timely manner; technology has had an immense impact on this process. With the advent of parallel machines, multi-processor scheduling problems have risen. Recently, broadcast scheduling problems have become relevant due to the way that sharing information has changed, including but not limited to satellite broadcast. Because of the pervasiveness of the Internet, algorithmic mechanism design for task scheduling on multiple (selfish) machines are being studied. As technological developments are constantly changing the relevance of scheduling models, understanding such problems from a theoretical perspective is the first step to solving these problems and increasing efficiency. Scheduling, in my opinion, is a prime example highlighting the contribution of theoretical research to society. Most real-world problems are so complex that they are intractable; without analyzing the issue in a simplified form, it is difficult to completely realize the full potential of technological impact on our society. This reason, coupled with the fact that I love studying these theoretical problems, is why I devote my time and energy to this field.

Although my most recent research activity has been primarily of an independent nature, I have also worked in groups to develop software and to present algorithmic results from a paper. I regularly study with others to improve on my weaknesses and to reinforce further my strengths by explaining concepts clearly to my peers.

I have served as an undergraduate teaching assistant for two semesters. One course was the third course in the introductory sequence for computer science majors. As it was programming intensive, it was also significantly more involved for me in my role as a teaching assistant, especially since I had never taken it before. In addition to learning the course material well enough to answer students' questions, I discovered that while it is not always easy to encourage a frustrated student, it is essential to that student's success.

The other course was tailored for non-majors, so I spent most of my time explaining the basic ideas of computer science to students with little or no computer science background. I constantly reminded myself that I take these simple concepts for granted; the 'array' was once foreign to me as well. Though the students may never look again at computer code, even basic concepts like looping structures can be applied to general problem solving. From an interpersonal perspective, this experience taught me how to relate to and explain computer science concepts to those who are not computer scientists. This skill is essential to effective communication on any level (for example, explaining theoretical research results to those who are not familiar with the problem and/or technique).

I am where I am now largely due to the fact teachers, mentors and academic role models along the way have encouraged me and created a supportive environment where I could learn and love computer science. I consider giving back to the community in this way imperative both to the advancement of the field and to helping minorities (women, for example) overcome the obstacles that others in the field do not face. At the same time, I have this strong interest in algorithmic design and analysis. The problems are very intriguing and span a multitude of fields, such as scheduling theory and economics. Therefore, it seems almost natural to pursue a position in academia, where both research and education are integral components.