I cannot understate the role teaching has had in shaping my career path thus far. Right from high school through the completion of my graduate program at Carnegie Mellon University, I have been fortunate to have had great teachers (and research advisers), especially for mathematics and theoretical computer science. Moreover, as short as my teaching stint has been (I have been a TA for two undergraduate courses at CMU), I have also felt the impact my teaching has had on the undergraduates. My passion for teaching stems primarily from witnessing, at both sides of the spectrum, the impact it can have on the students and their interests. In the following paragraphs, I will describe my philosophy towards teaching, how it has evolved, and my teaching interests.

Teaching Philosophy

Mechanics. I have served as a TA for two courses in CMU, an introductory course in theoretical computer science (Great Ideas of Theoretical Computer Science), and an advanced course on Probability and Computing. From my experiences in giving recitation lectures for these courses (and also giving research talks to general audiences), I have increasingly realized the importance of preparation. I vividly remember the discussion I had with a professor, after one of my worse talks, when he conveyed to me that every minute wasted is a minute wasted for each member of the audience. Ever since, I have always spent more time preparing my talks trying to make sure that I would not lose the audience. Additionally, I have realized that learning names of students and calling them by names during and outside lectures is an effective means of making them more attentive in class.

Ideology. It is imperative to choose the topics and even examples within each topic wisely, in order to retain the students' attention (especially undergraduates). For instance, in one of the probability recitations, we illustrated the use of Chernoff and union bounds with an example from discrepancy theory: given a set system, how do we color each point red or blue so that the discrepancy of any given set is small. While the students understood the solution, they did not seem very interested because we had not motivated discrepancy itself enough. In the next lecture, we decided to use a different example, showing what the NBA playoff system achieves: given n teams, create a tournament with $O(n)$ games so that the best team wins with constant probability. While this uses the same concepts, I felt that the students were more active because they could relate with the problem better.

I would also like to cover different topics in graduate classes each year. In my graduate years at CMU, Prof. Anupam Gupta offered four entirely different graduate courses on randomized algorithms, approximation algorithms, LPs and SDPs in algorithms, and advanced (exact) algorithms. It would benefit both the students, who get a good broad coverage of theoretical computer science, and the instructor, in keeping pace with recent developments in all areas of theoretical computer science.

Teaching Interests

My primary interests are broadly in topics in theoretical computer science, and related fields such as probability and discrete mathematics. For the undergraduates, I would like to teach an introductory course (like *Great Ideas in Theoretical Computer Science* at CMU), a course on *Algorithms and Data Structures*, and advanced elective courses on *Algorithms*, and *Probability and Computing*. The first introduces them to the large expanse that is modern-day theoretical computer science, and how it has really evolved by standing on the shoulders of giants. The second is more grounded, and introduces them to fundamental topics applied in all areas of computer science. The last two help prepare students first for undergraduate research and then for a graduate career in theoretical computer science. From my experience as a TA in CMU, I believe that a course focussed on the applicability of probability theory towards computing is an essential elective to offer, since probability is among the most widely used tools in all of computer science, let alone theoretical computer science.

As for graduate courses, I would like to teach specialized courses on *Approximation Algorithms*, *Randomization in Algorithms* (i.e., Probability and Computing), and *Combinatorial Optimization*. I would also really like to teach

courses related to my research interests, i.e., *Algorithm Design in the Face of Uncertainty*, and *The Algorithmics of Green Computing*. Finally, a course which I really enjoyed and intend to replicate, is a seminar course on *Theory's Greatest Hits* organized by Prof. Ryan O'Donnell. Such a course involves students volunteering to present seminal papers in theoretical computer science, spanning the last decade, and each student gets 2-3 hours to present the paper he has volunteered. In fact, one of the topics I remember the most, out of all the courses I have taken at CMU, is the topic I volunteered to present. Giving a two hour lecture on a paper to an audience really made me understand both the high-level ideas and the technical details thoroughly.

A general theme I intend to follow while teaching the above courses, is to place an emphasis on techniques rather than problems and solutions. This would convey the underlying ideas in a way befitting their applicability in other fields of computer science as well, making the courses suitable for a broad audience.