

# Teaching Statement

Prahladh Harsha

Teaching is an essential component of an academic life. I enjoy teaching and its thrills and challenges – interacting with students, attracting young minds to computer science, equipping them with a firm grasp of the fundamentals and disseminating recent developments in the field. As researchers it is often possible for us to get lost in the specific details of our work; teaching provides us with a broader outlook as it encourages us to constantly rethink our research in the general context of computer science and the real world.

**Teaching Experience:** I have had several teaching and mentoring opportunities with students at various levels, all of which I thoroughly enjoyed.

While a graduate student at MIT, I was a teaching assistant for six semesters, twice for the “Introduction to Algorithms (6.046)” course, once for the graduate course “Advanced Complexity Theory (6.841)” and three times for the “Theory of Computation (6.840)” course. The “Introduction to Algorithms (6.046)” and “Theory of Computation (6.840)” are large classes attended by a mix of about 120-160 undergraduates and graduates. My responsibilities for these two courses involved teaching a weekly recitation section attended by 15-20 students, holding office hours, assisting the instructor in preparing homework problems and exams, grading exams and supervising homework graders. Prof. Mike Sipser, the instructor for “Theory of Computation (6.840)” is an excellent teacher and has been consistently regarded by the students as one of the best lecturers at MIT. I learnt a lot TAing three times for his class. The other course I TAed at MIT was “Advanced Complexity (6.841)”, offered by Prof. Madhu Sudan. This was a smaller class attended by about 30 graduate students. My responsibilities for this course involved coming up with problem sets, grading the problem sets, holding office hours and maintaining the course webpage. I also substituted for the instructor for a couple of lectures, in which I taught topics from derandomization.

During one of the summers at MIT, I participated in a high school mentoring program. As part of this program, I would interact twice a week with two high school students. During each of these sessions, I would pose a challenge problem and via this problem introduce the students to problem-solving techniques and algorithmic ideas. It was a uniquely rewarding experience and I am very happy that one of the students went on to pursue a graduate program in theoretical computer science while the other is presently a senior in mathematics.

I had no teaching obligations during my research stints at Microsoft and Toyota Technological Institute; however, I offered two courses of my own interest: (a) a graduate course on expanders at Stanford<sup>1</sup>, which I co-taught with Dr. Cynthia Dwork (Microsoft Research) and (b) a graduate course on probabilistically checkable proofs at the University of Chicago<sup>2</sup>. The course on Expanders at Stanford was intended for graduate students doing research in algorithms and complexity, but I was pleasantly surprised to find students from other disciplines attending the lectures. Along with Dr. Dwork, I prepared the material for the course, gave 5 of the 10 two-hour lectures and prepared detailed lecture notes for the course (available online). I am pleased to note that these lecture notes have been used by other lecturers and students elsewhere. The course on probabilistically checkable proofs (PCPs) was an advanced graduate course with some of the most recent advancements in the area of PCPs and intended for graduate students doing research in complexity theory. I was gratified to see that graduate students from other backgrounds in computer science as well as a senior year undergraduate enroll for the course. Accordingly, I tailored the course to suit the students’ backgrounds while still conveying to them the recent, exciting developments in the field. The course involved some of the most technically involved topics in recent years in theoretical computer science. Despite this, the students actively participated in the classroom discussions, and with their help, I prepared detailed lecture notes for this material, which are now available online.

**Teaching Philosophy:** I believe a good teacher stimulates independent thinking in his/her students, while fostering an infectious enthusiasm in them. I realize that, as teachers, we can leave very strong impressions on students, thereby strongly influencing their academic and career choices. For example, my pursuit of mathematics and career-choices have largely been shaped due to the influential role played by my teachers since high-school. My primary goal as a teacher is to share my passion for my subject with my students, and reaching this goal can be very rewarding.

While teaching, I prefer the interactive style. I encourage active discussions both by posing questions myself as well as soliciting questions from the students during a lecture. Such a dialogue with the students helps me gauge the

---

<sup>1</sup>CS369E (Stanford, Spring 2005) – <http://ttic.uchicago.edu/~prahladh/teaching/05spring/>

<sup>2</sup>CMSC39600 (Univ. Chicago, Autumn 2007) – <http://ttic.uchicago.edu/~prahladh/teaching/07autumn/>

class and pace my lecture accordingly. Besides the traditional classroom discussions, I would encourage the use of online discussion groups and wikis to further student participation. Though these lack the personal one-on-one touch of a classroom, they offer the advantage of anonymity which would enable students, who are either naturally reserved or those from minority groups, to participate freely in the discussions. I also find the use of visual-aids (animations, pictures etc) while teaching very effective. Since my TA days at MIT, I have found it very useful to carry colored chalk/pens to lecture.

***Teaching Plans:*** I would love to teach popular computing courses on the lines of Steve Rudich's "Great Theoretical Ideas in Computer Science" at Carnegie Mellon and Sanjeev Arora's "Computational Universe" at Princeton. These courses go a long way in disambiguating the common misconception that computer science is merely the science of programming. I am excited to teach theoretical computer science courses, both introductory and advanced, such as theory of computation, algorithms as well as courses in the mathematical foundations of computer science. I am also interested in teaching introductory computer science courses where I would have the opportunity to attract young minds to computer science.