TEACHING STATEMENT

PEI WU

I consider teaching, or any educational practices in general, no less important than conducting research for various reasons. For the scientific reason, any field needs fresh blood to grow healthy. Most often, the "fresh blood" needs guidance in the beginning. For the sociological reason, teachers are what students see before they step into society. Being professional and serious about my work sets a good standard for students. For my personal reason, I really enjoy teaching. It is fun to share with students the knowledge that I enjoy (it is also fun to see students struggle as I used to). It is extremely satisfying to help students grow. Teaching is not unidirectional. Very often, students teaches me new perspective and new techniques. They are one strong force that motivates me to be a better person.

My expertise lies in theoretical computer science. My experience has prepared me to teach most theoretical courses at the introductory level (e.g. algorithms, automata theory), and some advanced courses (e.g. complexity theory). I plan to design more theoretical courses in the future as well. My top candidates include *quantum computing* and *analysis of Boolean functions*.

My teaching goals. I have three goals.

(i) I want to provide a solid foundational knowledge of the field. Students should understand what are the general questions in theoretical computer science, and why we study them. Students should master the tools that are widely used in theoretical computer science.

(ii) I want students to have exposure to research results beyond textbook. Our students should have bigger vision and think ahead. We don't need to go into any technical details if they are too advanced for the course. For example, when I teach randomized algorithms, I can mention that researchers believe randomness provides at most polynomial advantage because the research in *pseudo-randomness* has made remarkable progresses in reducing randomness...

(iii) I hope students to learn the skill of problem solving. A lot of students will be interested in other subjects and may never really need the exact course materials I teach. But the skill of problem solving can be helpful.

Ultimately, I want students to feel enlightened and accomplished. I want them to receive the best education that they deserve. I would be more than happy if someday some student runs into me and tells me with excitement that *because of my course*, he/she becomes a theoretical computer scientist (or anything).

My teaching strategies. To accomplish my aforementioned goals, there are a few teaching strategies that I apply. They are by no means exhaustive, but form a good collection for me to follow. The first four strategies focus on the procedure to present course materials in class. The fifth and sixth strategies emphasize that the overall lecture should get students involved and make them learn actively in class. The last strategy is for after class.

1. Provide background. When abstract definitions/theorems are introduced, it is sometimes not immediately obvious how these concepts are going to be useful. Without motivating these concepts properly, it is easy for students to lose their interest. Therefore I should provide background. For example, the first time when students encounter the complexity class polynomial hierarchy, they tend to wonder how is such thing useful anyway. I will explain to them. *Historically,* it is inspired by arithmetic hierarchy. It interpolates the more *familiar* NP and PSPACE, and is *equivalent* to Turing machines with access to a black box device.

2. Be precise and illustrative. In general, the materials I present should be definite, and precise. If I get sloppy, some students tend to be sloppy as well. Some other students will get confused and even struggled, then feel less interested. Both scenarios should be avoided. Other than being precise, the presentation needs to be illustrative as well, otherwise the course can be dull. I will constantly provide concrete examples. Colors and pictures are also very useful tools.

3. Pause every now and then. I pause after I introduce a new theorem so students have time to digest and even foresee the proof idea. I pause after I finish proving the theorem so students can ask for clarifications and check the steps.

4. Look back. After I complete a theorem (an algorithm or a section) in the lecture, I take the opportunity to look back at our proof. Can we simplify the proof? Is this condition necessary? Is our proof strategy useful for other problems? This is an important step that has many merits. It helps students to understand the result better, to appreciate it, and they are more likely to internalize the material.

5. Interact with students. I give eye contacts with students to check their status: Are they confused? Are they trying to think about the materials? I encourage students to share their own thoughts. I should be prepared for silence, and prepare hints. Students learns the best if they feel involved.

6. Ask questions. I ask questions and encourage students to ask questions as well. This is part of the previous item. However it is so important that I want to list it alone. First, asking questions is important for problem solving. Because, a general strategy to attack a new problem is to ask ourselves what problems/methods we know that are related to the current one. If we could not proceed, we ask ourselves can we simplify the problem somehow and solve the simpler version. During the lecture, I will explicitly and repetitively ask these questions. Students will naturally develop the habit when they are facing a problem on their own. In addition, our scientific progresses all start with questions. Students may feel hesitated in the beginning being afraid of feeling stupid. I can ask questions. Eventually they feel comfortable to ask their own questions.

7. Give homework and be ready to help. The only way to master any skill is to practice. I find it very helpful to mention homework problems in the end of the class. This enables students to connect their homework with specific course materials and motivates students to think about homework early. When students get stuck, it is important that they can find help. For this, I make sure students are able to access the course materials easily after lectures. In addition, I welcome face-to-face conversations with students during office hours. This way I can provide just enough hints for students to make progress. It also enables me to build more friendly relationship with students.