TEACHING PORTFOLIO

Abdelrahman Hosny

http://teaching.abdelrahmanhosny.me

Table of Contents

Cover Letter

November 10th, 2016
Computer Science and Engineering Department
University of Connecticut, Storrs, CT

Teaching in higher education has become very challenging. With more options available to the millennials, it became clear that teaching is not only the knowledge it leaves behind. Instructors have a bigger role of shaping their students' future. That's why I find extreme pleasure when asked to teach a class or deliver a presentation about my area of work. This teaching portfolio is currently not serving any certain application, but it is for my general curiosity to develop my career in ~~the~~ academia.

Enclosed is my teaching philosophy along with two courses I have assisted in teaching at the University of Connecticut. Although the content of the courses is mostly inspired by the current instructors whom I have worked with, I present my own strategy for delivering these courses in the future. Since I plan to adopt some parts of the courses for an online classroom, the document will receive several updates over the course of the next 12 months. Updates will adapt the course to an online course delivered through Udemy platform.

On that note, I'd be very grateful to receive any feedback or suggestions from you to launch the online course.

Sincerely,

Abdelrahman Hosny Ibrahim

Teaching Philosophy

There is a high probability that you are reading this document on a computer screen, a smartphone, or a tablet. Even if you are reading from a printed version, these words reached your hands through some kind of a computer. I am deeply passionate about computers and technology. I wish all people in the world could have access to the latest technology to improve their life. As Computer Science is leading the technological revolution, I think the global community should consider integrating Computer Science fundamentals curriculum in all majors. Learning Computer Science is not about classes. It is all about the learning environment. It is not a teaching-centric process, but a learning-centric one. Therefore, I believe that my role as a Computer Science teacher is mainly to support students during their learning journey of a specific topic. This support takes several forms from lecturing, organizing practice sessions, advising students and mentoring their careers. I divide the learning journey into three phases: Connecting, Learning and Applying, and I call it the CLA Journey.

In the *Connecting* phase, I do activities that prepare students to build up their knowledge based on their existing knowledge. Architects cannot build skyscrapers without a robust infrastructure and a floor-by-floor build process. They do not skip the seventh and eighth floors to build a ninth floor up on the air – all must be connected. Similarly, learning computer science builds up on accumulated knowledge. In general, a human brain cannot fully absorb new knowledge unless it is linked to some previous knowledge; that is how the neural cells work. So, in this phase, my goal is to link between what students know and what they are going to learn. As an example, surveying students' backgrounds through pre-class tests gives me a better idea on how diverse they are and allows me to introduce the class syllabus by reflecting on how the topics to be covered are related to their prior knowledge. Then, it is important to invest the first few classes in raising all students to the level of the class by discussing the prerequisites and ensuring that everyone has access to recommended readings. This phase ensures that students are ready to accept what is to be discussed in the material.

The *Learning* phase is when knowledge is actually built. I execute various forms of activities to support this phase. First, lecturing is where I transfer a capsule of experiences on a specific subject to students. Second, designing and running group projects provides a suitable atmosphere for students to reflect on the lectures and gives them a chance to rehearse by themselves. Third, organizing study groups fosters collaboration, which improve interpersonal skills. In a rapidly growing field like computer science, it is possible that what I teach in class won't be useful anymore sometime later. Hence, my focus is to improve their skills to learn any related topic on their own later. To check that they reached the goal of this phase, I flip roles. They illustrate what they learned by delivering a presentation, writing a blog posts, or using other media tools to communicate their ideas. The ground rule is: *If they cannot explain it simply, they do not understand it well enough*.

The brain's neural cells won't keep knowledge unless they are circulated in the brain by practicing. The *Applying* phase boosts this activity by letting students make use of the learned concepts in a real-world setting. Unless they get the chance to apply, their next step in their learning journey in other classes will be frustrating because there will be no hooks to build upon. My preferred direction in this phase is to adopt service learning. Students work on community projects that require the knowledge and skills they learned in class. For example, I ask them to develop a software for a charity organization. Not only do they use their knowledge, but also add a value to the community. This phase paves their road for future learning adventures.

Through the CLA journey, my supportive role for a healthy learning environment dominates my role as a teacher. It encourages students to enjoy learning, not just pass exams. A connecting phase prepares them for a class. A learning phase gives them a comfortable environment to learn new advancements in the field by themselves. Finally, an Applying phase gives an opportunity for students to reflect on their gained knowledge and make a real impact on the community. Computer Science cannot be taught, rather it should be learned and my role as a teacher is to facilitate this learning journey.

Algorithms and Complexity Course

Algorithms and Complexity Course

Syllabus

Course goals and learning objectives:

Algorithms are the heart of computer science, and the subject has countless practical applications as well as intellectual depth. This course is an introduction to algorithms for students in computer science majors. The course emphasizes the mathematical and conceptual understanding over the low-level implementation. After completing this course, you will be well-positioned to:

1. Solve common computational problems in computer science.
2. Analyze the time complexity of a given solution.
3. Identify best strategies to solve a given computational problem.
4. Come up with a complete and efficient solution for a given computational problem.

Course Prerequisites

- Some programming experience. In particular, you should understand recursive procedures and simple data structures such as arrays and linked lists.
- You should have some facility with mathematical proofs. A few portions of the book rely on some knowledge of elementary calculus.

Topics to be covered

- Preliminaries, Sorting and Selection
- Divide and Conquer
- Important Data Structures
- Dynamic Programming
- Greedy Algorithms
- Graph Algorithms
- Parallel and Randomized Algorithms
- Intractable Problems

Textbook

Introduction to Algorithms by T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, Third Edition, MIT Press, 2009.

Additional References

The topic of algorithms continues to grow rapidly and we cannot cover all aspects of the topic. It is highly encouraged to stay on top of the news by following up with the recent advancements in the topic. Since textbooks takes time for publication, we highly recommend to search online for the updates on each topic covered in class. Bring the latest update you have and we will discuss it briefly in class.

Grading

- 30% for homework (10% for each; 3 homework in total)
- 30% for the midterm exam.
- 40% for the final exam.

Homework is to be delivered in class. You can deliver your homework late up to 2 days loosing 20% of your grade on that homework. No homework is accepted after that.

Communication

There are no office hours. If you see my office door open, that means I'm ready to let you in to talk. Come visit me anytime. Have a busy schedule? Email me to confirm a meeting time.

Course Policies and Academic Integrity

Discussion with classmates is encouraged for homework problems. However, you must deliver your homework on your own. Do not copy any other person's work or online solutions. Use appropriate references if your solution is inspired by some online resource.

We will follow the University Policy on Academic Integrity regarding any cheating and plagiarism. Take the time to familiarize yourself with the contents of this page, as you are responsible for its contents.

Weekly Schedule

Week	Topics	CLA Phase	Notes
1	Preliminaries	Connecting	
2	Sorting	Learning	
3	Selection	Learning	HW 1 posted
4	Important Data Structures	Learning	
5	Divide and Conquer	Learning	HW 1 due
6	Cont. Divide and Conquer	Applying	HW 2 posted
7	Greedy Algorithms	Learning	
8	Cont. Greedy Algorithms	Applying	HW 2 due
9	Dynamic Programming	Learning	Midterm (outside class)
10	Cont. Dynamic Programming	Applying	
11	Graph Algorithms	Learning	HW 3 posted
12	Cont. Graph Algorithms	Applying	
13	Parallel and Randomized Algorithms	Learning	HW 3 due
14	Intractable Algorithms,	Learning, Applying	Extra exercises posted
15			Final Exam

Assignments

Homework	Measures
1	a. Employing preliminary mathematical formulas in proper domains. b. The ability to analyze the run time of a given algorithm. c. Use different sorting algorithms and compare between their run times. d. Use different selection algorithms and compare between their run times. e. Proper use of studied data structures.
2	a. The ability to design algorithms using the divide-and-conquer paradigm. b. The ability to think greedily in problem solving.
3	a. The ability to design algorithms using the greedy algorithms approach. b. The ability to design algorithms using the dynamic programming approach.

	c. Familiarity with graph algorithms and their algorithms. d. The ability to design algorithms that work on graphs.

Exams

Exam	**Measures**
Midterm	a. Overall understanding of why we need algorithms in computer systems. b. The ability to analyze the run time and disk space requirements of a given algorithm. c. Solve problems by designing algorithms inspired by topics discussed in class. d. Employ data structures in their solutions.
Final	a. The ability to classify a given solution of an algorithm under the categories studied. b. The ability to come up with a solution to problems not covered in class, using the techniques studied throughout the semester.

Case Studies

In the *Algorithms and Complexity* course, case studies become very useful in applying an integrative study on common problems in computer that require interdisciplinary techniques to solve. Case studies are not graded. They are posted along with their candidate solutions for further readings by motivated students. There will be two presented case studies that introduce students to real-world problems.

Case Study 1: File Indexing

The goal is to Index a PC (or the web). Given two or more words, report the list of (text) files in which these words occur. Students should tackle the following issues:

- What data structure(s) to use.
- Which pieces of information to save.
- How to perform a search given the keywords.
- The performance of the given integrated solution.
- Question to ponder: what if data is so large that it does not fit in RAM?

Case Study 2: Travelling Salesman

The goal is to design a plan for a salesman to visit a number of cities to sell his company's products. The trip should visit all cities with the minimum travel cost. It is known that the best known solution for this problem takes exponential time to compute, which is inefficient. Students are introduced to this case study to tease their desire to discover a broader range of intractable problems applications.

- How to represent the problem as a graph.
- How to model travel cost (distance, flight, accommodation, etc.)
- Question to ponder: can we come with a "good" solution that is not perfect, but takes less time?

Evaluation

To be filled out by students.

1. What is the overall rating?
    - Excellent
    - Very Good
    - Good
    - Fair
    - Poor
2. Indicate your level of agreement with the following statements:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The course objectives were clear					
The course textbooks were clear and well written					
The assignments were appropriate for the level of this class					
The course increased my interest in the subject					
The course corresponded to my expectations					

3. Please provide any comments or suggestions that might help improve this course in the future?

Statistical Analysis for Computer Systems Course

Statistical Analysis for Computer Systems Course

Syllabus

Course goals and learning objectives:

Probabilistic models underlie important applications computer science. This course is an introduction to the probability theory and statistical analysis for students in computer science majors. The course establishes the mathematical understanding and connects it to case studies in computer science. After completing this course, you will be well-positioned to:

1. Define probabilistic models and differentiate between them and the deterministic models.
2. Understand the probability theory.
3. Apply discrete and continuous random variables in analyzing different systems.
4. Identify common distributions (Binomial, Poisson, ...etc.)
5. Establish a statistical hypothesis and test it.

Course Prerequisites

- Elementary calculus is a must.
- Programming knowledge in any language is required for the class project.

Topics to be covered

- Sample space and events
- Discrete random variables
- Continuous random variables
- Expectation and moments
- Stochastic processes
- Discrete Time Markov Chains (DTMC)
- Statistical inference
- Continuous Time Markov Chains (CTMC)

Textbook

K. S. Trivedi, Probability and Statistics with Reliability, Queuing and Computer Science Applications, Second Edition, John Wiley.

Grading

- 30% for projects (2 projects; 15% for each)
- 30% for the midterm exam.
- 40% for the final exam.

Projects are to be delivered online through Blackboard system. You can deliver your project late up to 2 days loosing 20% of your grade on that project. No deliveries are accepted after that.

Communication

There are no office hours. If you see my office door open, that means I'm ready to let you in to talk. Come visit me anytime. Have a busy schedule? Email me to confirm a meeting time.

Course Policies and Academic Integrity

Discussion with classmates is encouraged for project issues. However, you must deliver your project on your own team. Do not copy any other team's work or blindly copy source codes from online repositories. Use appropriate references if your solution is inspired by some online resource.

We will follow the University Policy on Academic Integrity regarding any cheating and plagiarism. Take the time to familiarize yourself with the contents of this page, as you are responsible for its contents.

Weekly Schedule

Week	Topics	CLA Phase	Notes
1	Preliminaries	Connecting	
2	Sample space and events	Learning	
3	Discrete random variables	Learning	
4	Continuous random variables	Learning	Project 1 posted
5	Applications on random variables	Applying	
6	Expectation and moments	Learning	
7	Stochastic processes	Learning	Project 1 due
8	Cont. stochastic processes	Learning	Midterm
9	Case studies	Applying	
10	DTMC	Learning	Project 2 posted
11	CTMC	Learning	
12	Applications on Markov chains	Applying	
13	Statistical inference	Learning	Project 2 due
14	Reflection on probabilistic vs. deterministic models	Learning, Applying	
15			Final Exam

Group Projects

Exam	**Measures**
Project 1	a. The goal is to practice calculating basic probability metrics (mean, median, variance, standard deviation). b. The project focuses on simulating random walks in a two-dimensional grid. Students are asked to collect data as the simulation runs and then calculate the required metrics.
Project 2	a. The goal is to implement Google's PageRank algorithm and understand the stochastic process in its core. b. The project focuses on the implementation and observation of how stochastic processes run in practice. c. This project includes a research part as it motivates students to integrate their class material with the state-of-the-art enhancements of the concepts.

Exams

Exam	**Measures**
Midterm	a. Overall understanding of probabilistic models. b. The ability to calculate basic probability metrics. c. Solve problems on discrete and continuous random variables. d. Solve problems on expectation and moments.
Final	a. Overall understanding of probabilistic versus deterministic models. b. Solve problems on stochastic processes. c. Solve problems on Markov chains. d. The ability to state a hypothesis and prove or disprove it using statistical inference.

Evaluation

To be filled out by students.

1. What is the overall rating?
    - Excellent
    - Very Good
    - Good
    - Fair
    - Poor
2. Indicate your level of agreement with the following statements:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The course objectives were clear					
The course textbooks were clear and well written					
The assignments were appropriate for the level of this class					
The course increased my interest in the subject					
The course corresponded to my expectations					

3. Please provide any comments or suggestions that might help improve this course in the future?