EECS 2011 Week 1

Ilir Dema

Today's agenda

- **Why** learning EECS 2011
- **What** we'll learn in EECS 2011
- **How** to do well in EECS 2011

Why EECS 2011?

To nail job interviews!

Scenario: The Tech Interview

- **Interviewer:** You are given a set of courses where each course stores a list of prerequisites. Devise an algorithm that returns a valid ordering of taking these courses.
- **You**: (think for 1 minute) Here is my algorithm:
  - For a valid ordering to exist, there must be a course **X** that has no prerequisite.
  - I choose **X** first, remove **X** from the set of courses, then remove **X** from all other courses' prerequisite list.
  - Find the next course in the set that has no prerequisite.
  - Repeat this until …
- **Interviewer**: OK stop talking that's it for today thank you for your interest and we'll call you about next steps.
- **You**: But you don't have my number…
- **Interviewer**: Next one!

Scenario: The Tech Interview, Take 2

- **Interviewer:** You are given a set of courses where each course stores a list of prerequisites. Devise an algorithm that returns a valid ordering of taking these courses.
- **You**: This is a **topological sort** problem which can be solved using **DFS**.
- **Interviewer**: YOU GOT IT!

EECS 2011: Fundamentals of Data Structures

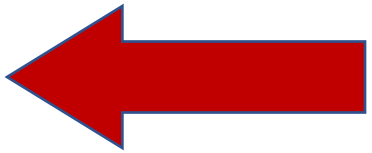
- What you'll learn in this course are some **FOUNDATIONAL** skills needed in almost every field of Computer Science and Engineering.

- You will design data structures and algorithms like a **PRO**.

What's in EECS 2011?

Fundamentals of Data Structures

Data Structures



and

Analysis

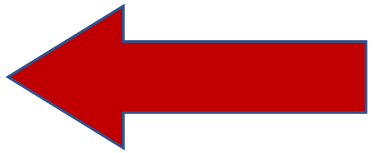
Data Structures

- Linked Lists
- Stacks and Queues
- Trees, Binary Trees, Binary Search Trees, Balanced Binary Search Trees
- Heaps
- Hash Tables
- Graphs
- Recursion, Sorting
- …
- We learn to identify the **structure of a problem** and apply the appropriate design technique to devise the **optimal solution**.

Fundamentals of Data Structures

Data Structures

and

Analysis 

Analyzing Algorithms and Problems

Reasoning about algorithms and problems

- Prove the **correctness** of an algorithm.
- Prove the **complexity** of an algorithm.
- Prove that a problem **cannot be solved** with less than X amount of time.
- Prove that a problem **cannot be solved by a computer**.

- **… math and proofs**

But **analysis** is really more important…
Think about cooking…

- A chef can **analyze/reason** about the effect of every ingredient on the dish's flavour, so a chef can invent new dishes.

- Only when you know how to **analyze/reason**, you can invent your own new data structures rather than simply repeating existing ones.

EECS 2011 does NOT just turn you into a better programmer.

It turns you into a better **computer scientist**.

Math background needed

- Pre-condition, post-condition
- Logic
- Proof by induction, proof by contradiction
- Summation of series
- Counting and probability
- Elementary graph theory

How to do well in EECS 2011

First and Foremost

Be Interested.

Course Website

- Everything is on eClass
- (Let's take a look at the eClass page)

Course Syllabus

(Let’s go over the course syllabus on eClass)

Please read it thoroughly. Very important!

A Tip for Lectures

- Get involved in classroom **interactions**
  - asking/answering a question
  - discuss with the class
  - making a guess / bet / vote
  - actively work on the exercise
- **Emotional involvement makes the brain remember better!**

Checklist: How to do well

- Be interested.
- Check course page and course announcements daily.
- Attend lectures and actively interact.
- Work on practice problems and finish on time.
- Review lecture slides.
- Read the textbook for more info.
- Discuss on the discussion board.
- Go to office hours.
- Work hard on assignments and submit on time.
- Give feedbacks.
- Ask the instructor for help when needed. (**Use the One-on-One Chat. No emails.**)
- Do well in tests.

Next class

- Start learning!
- Linked Lists