ALGORITHMS
Advices & Syllabus

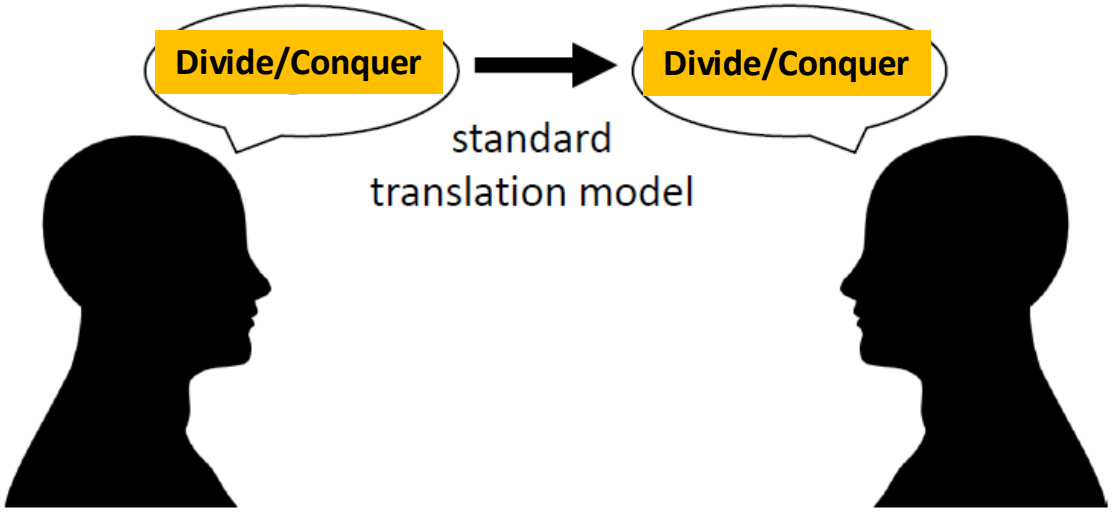
Teaching/learning/understanding dilemma

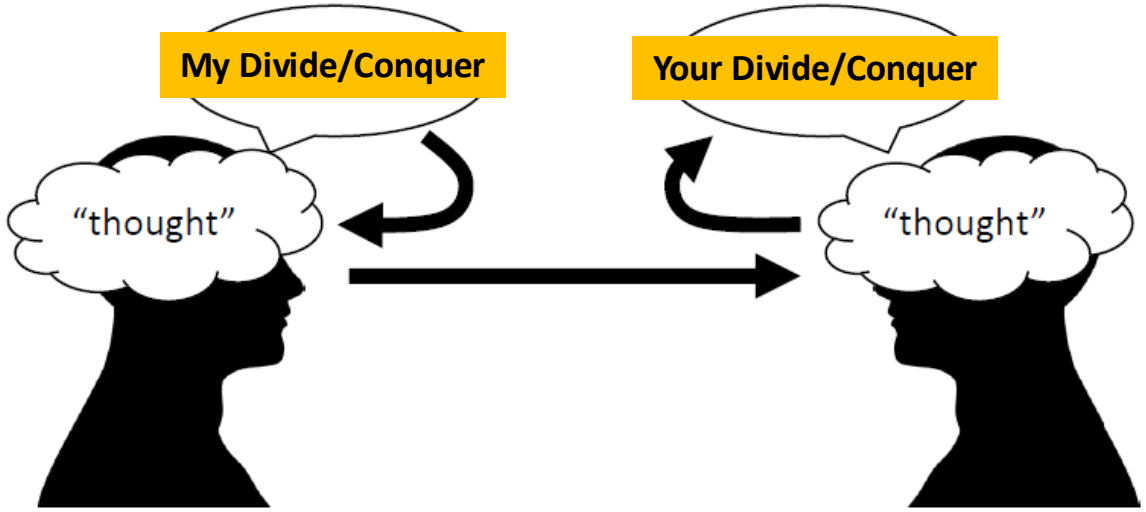
- At university level, a lecturer's job is to show you the possibilities and some resources (slides, exercises, books, videos etc.)
  - Instructor in/teacher out
- Your job is to **teach yourself** the lectures.
  - You are the teacher of your own brain.
  - My goal is to challenge you. Brain is lazy when it feels no danger.

Teaching/learning/understanding dilemma

- If you think we can make you learn the courses, you are grossly overestimating our abilities.
  - We don’t have telepathic powers.
  - Being taught != learning != understanding/knowing
  - “I learned very early the difference between knowing the name of something and knowing something.”
    — Richard P. Feynman
- If you think you can understand topics with little individual effort, most probably, taking that course makes no sense at all.
  - Develop ur own metrics to evaluate yourself on how well you have learned.
    - Barrier to entry
    - The more you understand, the more enjoyment you feel?
  - Your effort depends on your **background**. Take ur own responsibility. Oftenly when we fail, we try to prove ourselves right. We look for evidence to support our ego.

What is going on?



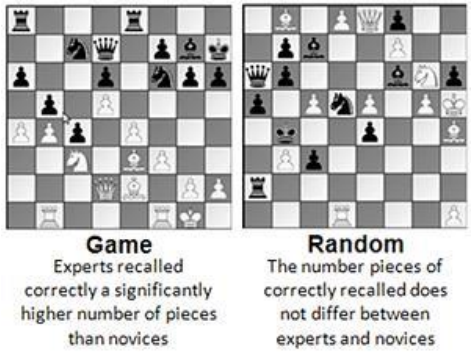


the "thought" is a **representation**!

- Different backgrounds, different ability levels, different interests, different motivations etc.
- There is no universal learning strategy.

Some hints

Chase&Simon, 1976



- Experience: The brain will find connections to previous lectures, textbook content, and personal life experiences, which serve as the foundation in understanding.
  - We store information in terms of meaning not the sensors.
  - Am I ready to understand this?
  - We tend to look for information that fits to our knowledge and we ignore that doesn't fit.
- Self-Explanation/curiousity: How do I know? Why does it make sense that? Why is this true?
- You should be studying regularly instead of cramming for exams so that enough time is dedicated to each subject before its exam.

Some pitfalls

- No one gains program solving skills by listening passively.
  - Requires disciplined activity and acquisition.
- Just because you wrote "something", it doesn't mean you will get any partial credit if your answer has nothing to do with the question.
- Sometimes, we fool yourselves by saying “I got the answer from a friend but he explained it to me, and I understood it!”
  - Undertanding doesn’t work that way. That’s why they say “no pain, no gain!”. The amount of understanding that takes place is proportional to the amount of work/pain involved. Understanding takes place through practice as the brain is rewired during the process, which is a time-consuming one.

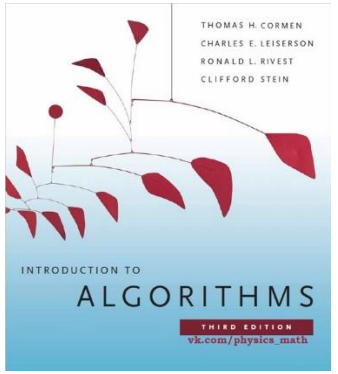
Who is this course for?

- In big company interviews, they will not ask you to make an endpoint using Express/Spring.
- Algorithms/Computer science theory
  - We want you to write a code for solving this problem.
    - Not straightforward. Requires algorithmic/analytical thinking.
  - Nice, but we have some restrictions:
    - Efficiency (time complexity) o.w. time-out error.
    - By watching Youtube videos, you can develop Amazon from scratch without touching this issue.
    - There are edge test cases (Think the unthought).
- Get ready to see puzzle-like questions.
  - Don’t say "Hocam, these questions did not seem familiar with the lectures."
  - You will find patterns if you gain experience.

Prerequisities

- Data structures
- Honesty to yourself
- Patience

Resources



- T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to Algorithms", MIT Press and McGraw-Hill, 2009. Third Edition
- Internet (e.g. geeks4geeks.com, leetcode.com)
- Youtube lectures (e.g. Abdul Bari)
- Other universities' Algorithms exams

Outline

Concepts	Textbook
Introduction to Analysis of Algorithms	Chapters 1 & 2
Asymptotic Notation	Chapter 3
Solving Recurrences	Chapter 4
MWE-1 (10%)	
The Divide-and-Conquer Design Paradigm	Chapter 2 & 4
Quicksort	Chapter 7
Randomized QuickSort	Chapter 7
Randomized QuickSort Cntd	Chapter 7
MWE-2 (10%)	
Median and Order Statistics	Chapter 9
Midterm (20%)	
Heapsort	Chapter 6
Dynamic Programming	Chapter 15
MWE-3 (10%)	
Greedy Algorithms	Chapter 16
HW-out (10%)	
Huffman Codes	Chapter 16
HW-in	
Final (40%)	

Grading

Component	Weight (%)	Date/Due (Approx.)
5 Hackathons	40	Weeks 3, 5, 7, 10, 12
Midterm	20	Week 8
Final	40	

- Always keep in mind that both you yourself and I want you to be successful; hence we are on the same side!