Intro to CS16

CS16: Introduction to Algorithms & Data Structures
Spring 2019

Welcome to CS16!

What is 16 about?

Algorithms

“sequence of computer instructions for a given task”

Why Study Algorithms?

- Core to Computer Science
- Powerful in practice
  - running in seconds vs. age of the universe
- Interesting and elegant ideas

Web Search in the 90’s

Google

- My friend Jeremy tells me about this new engine
  - "it's awesome you should try it out!"
- After `10` minutes it's obvious that
  - Google results were more "relevant"
- But why?
  - Why were Google's results better?
  - What was Google's secret?
- I finally learned why during my PhD
  - Google had a **better algorithm**!

Google

- Before Google
  - search engines ranked pages using keyword frequency
  - well-known and worked OK
- Larry Page & Sergey Brin (PhD students @ Stanford)
  - noticed that links were important too!
  - intuition that links conveyed information about importance
  - But what exactly? and how can you make use of it?
  - Lead them to design the PageRank algorithm

PageRank

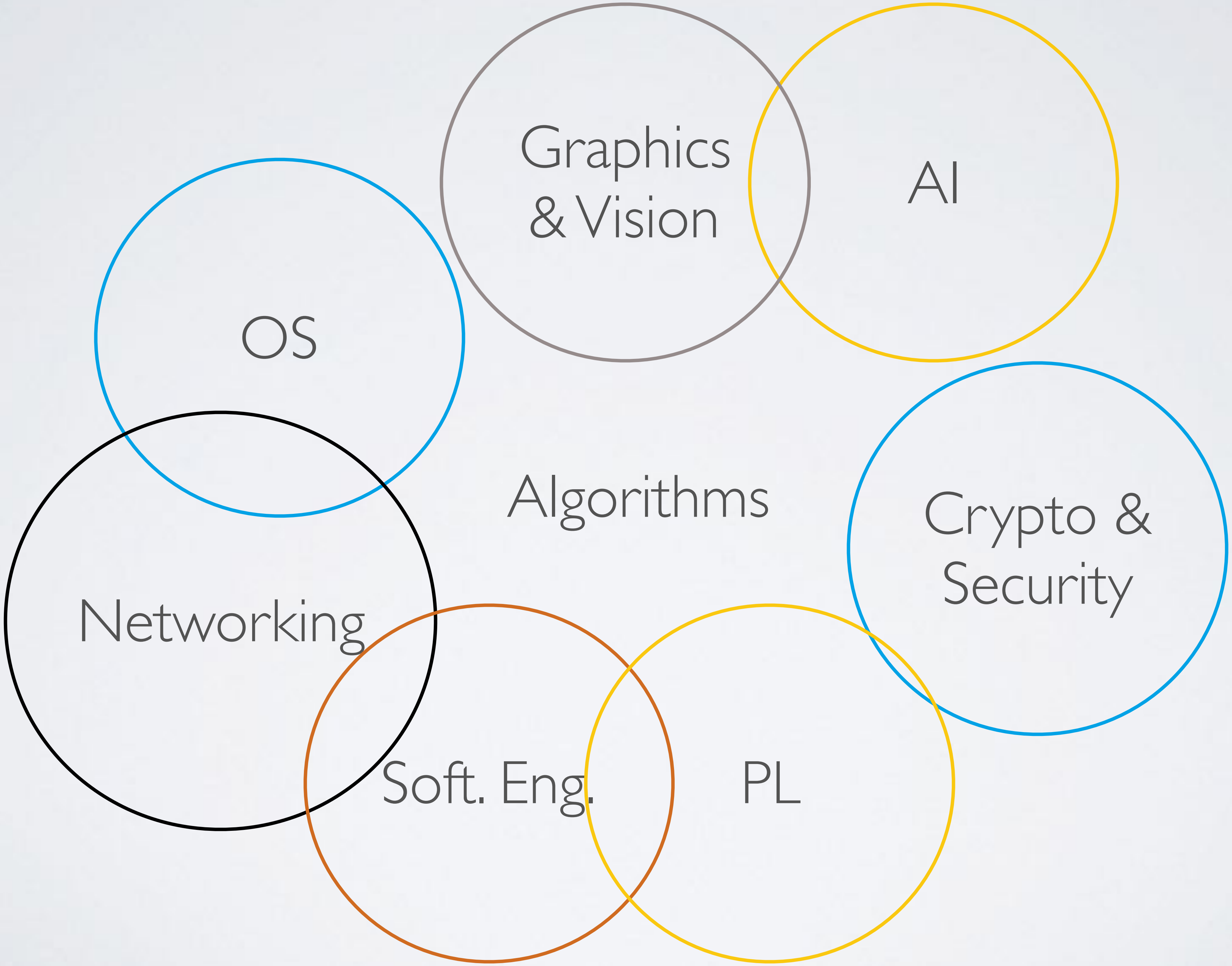
n32
n23
n16
n13
n14
n21
n12
n36
n24
n10
n8
n25
n3
n20
n35
n2
n11
n19
n0
n7
n1
n22
n34
n15
n27
n39
n9
n18
n17

- How does PageRank work?
- Why does it work?
- How do you implement it efficiently?
  - Google indexes "hundreds of billions" of pages
  - answers and ranks in 0.5 seconds
  - processes 40,000 queries a second
  - 3.5 billion per day
- Using clever **algorithms** and **data structures**!

A Personal Example

- Searching on encrypted data
  - Really cool idea; but slow
- We thought about this a lot
  - new approach that was very fast
  - as fast as searching on unencrypted data!
- 15 years later & after a lot more research
  - startups, apps, real-world products
- What was the secret?
  - clever use of **algorithms** & **data structures**

CS is diverse



You Will Learn

- How to **design** algorithms
- How to **analyze** algorithms
- How to **implement** algorithms

Content of CS16

- **Analysis of algorithms:** big-Oh, worst-case analysis, amortized analysis, expected running time
- **Design paradigms:** dynamic programming, divide and conquer, greedy algorithms
- **Recursive algorithms:** recurrence relations, induction
- **Elementary data structures:** stacks, queues, trees, hash tables, binary search trees, heaps, graphs
- **Sorting algorithms:** insertion sort, selection sort, heap sort, merge sort, quicksort, radix sort
- **Machine learning algorithms:** decision trees
- **Graph algorithms:** depth-first search, breadth-first search, shortest path, minimum spanning tree, topological sort
- **Advanced topics:** Bitcoin, functional programming, numerical algorithms

Meet your TA’s

Goals

- *Learn* fundamental algorithms and data structures
- Find and *design* new ones
- *Reason* about them
- *Use* them
- *Prepare* you for more CS

© AFP/Getty Images

Lectures

- Cover various algorithms & data structures
    - How they work
    - Why they work
    - Analyze them
- Activities & discussions
- You are responsible for content in lecture (whether on slides or not)

Textbook

- No required textbook
- Helpful
  - Dasgupta, Papadimtriou and Vazirani
  - Goodrich and Tamassia
  - Roughgarden

Course Page

- Slides
- Notes
- Announcements
- Helpful readings

Piazza

- Announcements
- Questions and answers
- Links to helpful material (blogs, Youtube videos)

Office Hours

- TA office hours are very helpful
  - Try to get unstuck on your own first
- My office hours
  - Mondays 4-6
- Questions about HW or projects:
  - Post on Piazza
  - Ask in Section
  - TA office hours
  - Schedule meeting with me

Homeworks (30%)

- `10` HWs
- Due every week
- Python code, proofs, analysis, …

Projects (30%)

- 4 projects in Java

Sections (10%)

- 1 hour/week with 2 Tas
- 6-10 students
- Required!
  - If you miss 3 you fail
  - Lose points for every missed section
- Mini assignments
- Mentor

Exams (25%)

- Midterm
  - Date: March 21st
- Final
  - Date: May 14th

Collaboration

- Encouraged to collaborate on HWs but
  - **Write up HWs by yourself**
  - **Code by yourself**
  - **No sharing of code or pseudocode**
- **No collaboration on Projects**
- You will sign the collaboration policy
- We will use code similarity tester
- Random live audits
  - might ask you "what would happen if we did X to your code?"

Override Policy

- Email HTAs if
  - You are a graduate student
  - You are RISD student and need a signature

Email Policy

- Unless matter is private always email HTAs!
  - Your email can get lost in my inbox
  - It may take me a while to get to your email
  - HTAs may get to it faster & will remind me

References

- Slide #2
  - A statue of Muhammad ibn Musa al-Khwarizmi; a persian scholar from the 9th century
  - “Algorithms” is derived from “Algoritmi” which is the Latin translation of his name
  - Worked in mathematics, astronomy and geometry
  - Founded the field of Algebra
- Slide #10
  - Lionel Messi is a soccer player that plays for Barcelona and Argentina
  - He is considered one of the best soccer players of all time
  - Won 5 Ballon D’ Ors, 8 La Liga titles, 4 Champions League titles
  - Scored the most goals and made the most assists in La Liga history