

# CSE 91: What did we do and where do you go from here?

CSE 91 Last Lecture

George Varghese

# CSE 91 Goal 1: Essence

- 1) To convince you that Computer Science is not just programming but **creatively** solving the world's **problems** using **computers**
- **Spam** (Savage): **understanding money chain**
  - **Pollution** (Griswold): **network of cheap cell phone sensors**
  - **Elder Care** (Ettinger): **Automatic cameraman**
  - **Online Cosmetics** (Kriegman): **Makeover site**
  - **Buying rare stuff** (Elkan): **Recommendation via learning**
  - **Cancer detection** (Bafna): **Placing primers optimally**
  - **Underwater sensors** (Kastner): **Underwater modems**
  - **Web sites for masses** (Yannis): **Automatic DB schemas**

# Enabling technologies for solving problems with computers

- Programming hard (Rondon): Program on web
- Security poor (Savage, Bellare): Public key crypto
- Stuff Fails (Marzullo): Byzantine generals
- Too much Power (Agrawal): Green computing: Cell2Notify
- Interact with world (Gupta): Embedded systems: Boards
- Too slow (Dasgupta): Algorithms: Dijkstra's
- Often buggy (Graham): Discrete math: Probability trees
- Need platforms (Varghese): HW, Arch, OS: Time sharing

# Common theme: abstractions

## End User Abstractions

- Programming abstractions (Rondon): **circle**
- Face makeover abstractions (Kriegman): **PaintLip**
- Camera operating commands (Ettinger): **Clap**
- Recommendation systems (Elkan): **NetFlix Recommends**
- Search (Varghese): **Google box**

## Programmer Abstractions

- Abstract Data Types (Dasgupta): Graphs, **shortest paths**
- Operating System: (Varghese) Virtual Memory, **Windows API**
- Architecture: **x86 Instruction Set** for OS programmer

# So what is Computer Science?

- Study of **abstractions** to enable people to creatively solve the world's problems and to build better tools for programmers.
- Also, effort to make these abstractions more **useful** (secure, reliable, correct, efficient, low power)
- Also a **way of thinking** (e.g., algorithm, program, virtual) that can transform other fields: math, science, social studies
- Programming is how we **use** the abstractions but gives you no insight on how to **build** better abstractions!
- This is why its worth taking Undergraduate CSE courses

# CSE 91 Goal 2: Startup exposure

- 2) **Startups:** To give you a glimpse of how CSE ideas can convert to business opportunities
- **CSE91 Examples:**
  - Taaz (Kriegman), finding right business model
  - NetSift (Varghese), startup story
  - Google, ad words business model
  - App2You (Yannis), filling a need
  - Homework on business models and cap table

# CSE 91 Goal 3: Showcase UCSD CSE

- **UCSD CSE:** To show you that UCSD CSE has a number of cool professors doing cool work
  - **Kriegman:** interacts with fashion designers
  - **Savage:** works with Interpol on Internet Security
  - **Bellare:** helped design modern crypto systems
  - **Elkan:** helped design Netflix Prize contest
  - **Marzullo:** helped design NTP, plays ukelele
  - **Graham:** won Steele Prize in math, juggler
  - **Kastner:** dives in Tahiti to do his research
  - Lots of other award winning faculty (Oscars etc!)

# CSE 91 Goal 4, Students like you

- **Students:** To showcase students like you:
  - 3 lectures by students (Rondon, Ettinger, Agrawal)
  - Satya Mallick (co-founder of Taaz)
  - Sumeet Singh (co-founder of NetSift)
  - Keith Kowalczykowski (President, App2You)
  - Netflix prize idea due to UCSD student
  - Savage, Griswold, Kastner, Gupta all work with undergraduates

# CSE 91 Goal 5, Career Opportunities

- **Careers:** To show there are exciting career options that can change the world
  - Build great products at Google, Microsoft etc.
  - Do interdisciplinary work (Biology, Cog Science)
  - Start a Company
  - Help some world cause
  - Be researcher or work on cutting edge products

# Option 1, Build Great Products

- Possible next steps:
  - Become a tutor for a programming course
  - Join programming contests
  - Take the Games Class and the Rendering Class
  - Learn useful technologies (e.g., learning theory, algorithms, hardware, vision) you need
  - Do internships early

# Option 1, Example



- Lindsey De Salvo: UCSD CSE 2005
- Software engineer, Amazon
- Academic coordinator Tau Beta Pi, volunteered for San Diego Rescue Mission

# Option 2, Interdisciplinary Work

- Possible next steps: (e.g., CSE and Theater)
  - Consider double majoring and using BA in CS
  - Check out programs in Bioengineering and User Interfaces from Cognitive Science
  - Check out opportunities at CALIT next door
  - Learn stuff in other area, take courses
  - Do internships in other area

# Option 2 Example



- Khwaja Shams: UCSD CSE 2006
- Works at NASA Jet Propulsion Lab
- Software for rovers and landers on MARS

# Option 3, Start a Company

- Possible next steps:
  - Check out opportunities and courses at Von Liebig center
  - Try out entrepreneurship challenge (OK to fail)
  - Check out opportunities at Rady School
  - Learn or read books on startups, accounting, sales
  - Do internships at a startup (Taaz had one)
  - Make allies with folks with complementary skills

# Option 4, Example

- **Example:** Nadav BenHaim, co-founded license recognition startup after undergrad with UCSD Vision Professor Serge Belongies

# Option 4, Help the World

- Possible next steps:
  - Find some cause you really care about
  - Figure out how computers can help cause
  - Try to meet people working in this cause and find what the real needs are.
  - Take field trips (e.g., to Africa)
  - Write a tool for cause as a prototype

# Option 5, Do Cutting Edge Research

- Possible next steps:
  - Find a research opportunity at department, CALIT, SDSC, anybody who will take you
  - Try to do well at some courses (projects, extra mile) so they will write you recco letters
  - Try to write a paper
  - Try and find an area you are passionate about
  - Apply to Grad School, get an MS or PhD

# Option 5 Example



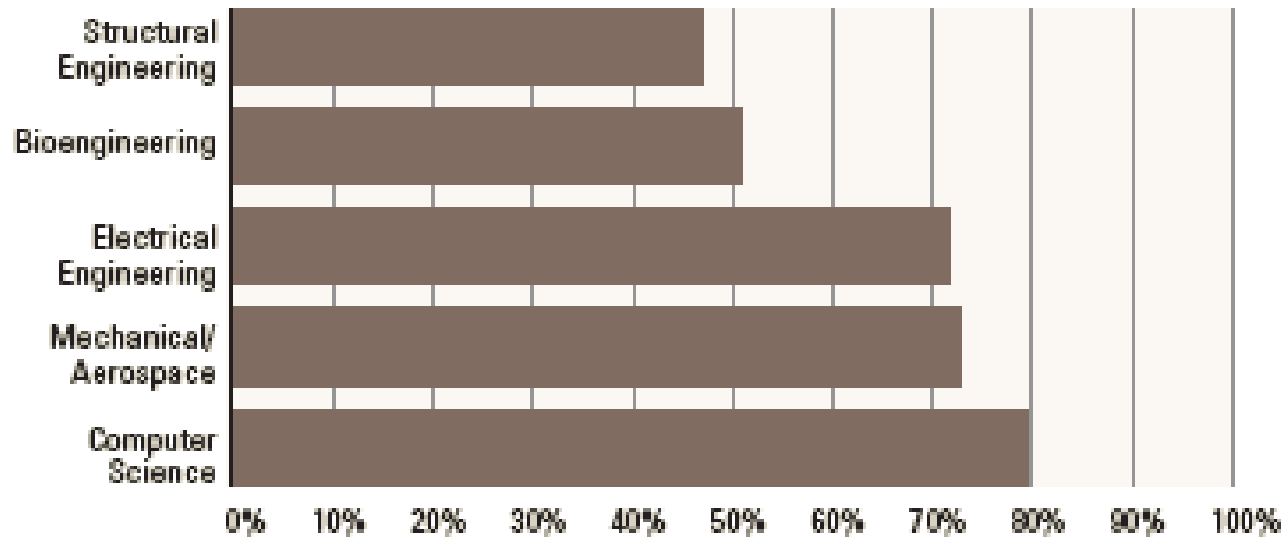
- Angelina Lee: UCSD CSE 2003
- Ph.D. candidate at MIT under Charles Leiserson
- Parallel processing that can handle errors well

# CSE Jobs

- So its exciting , but are there good jobs? Yes!
  - 80,000 new CSE jobs in California, more than any other degree (CA Labor Market Analysis 2007)
  - Highest growing market nationally (25%) (US Bureau of Labor Statistics, Nov 2007)
  - 59% of all new job openings 2004-2014 vs life science 4% (US Bureau of Labor, 2005)
  - Median salary \$70,000 (labor statistics) 2<sup>nd</sup> highest
  - Unemployment 3% is low versus US average 6%.  
91% of employers report shortage

# UCSD Engineering Starting Salaries

**2007 Starting Salaries  
Greater Than \$56,000**



# Graduate School: Why, how and when

Stefan Savage  
UC San Diego



# Really?

- I've been in school for 13 years so far...  
• (K-12.)
- I have another four to go... (BS/BA)
- And you want me to spend another:
  - 2-3 years (M.S.)
  - 5-6 years (Ph.D.)
- No, really?



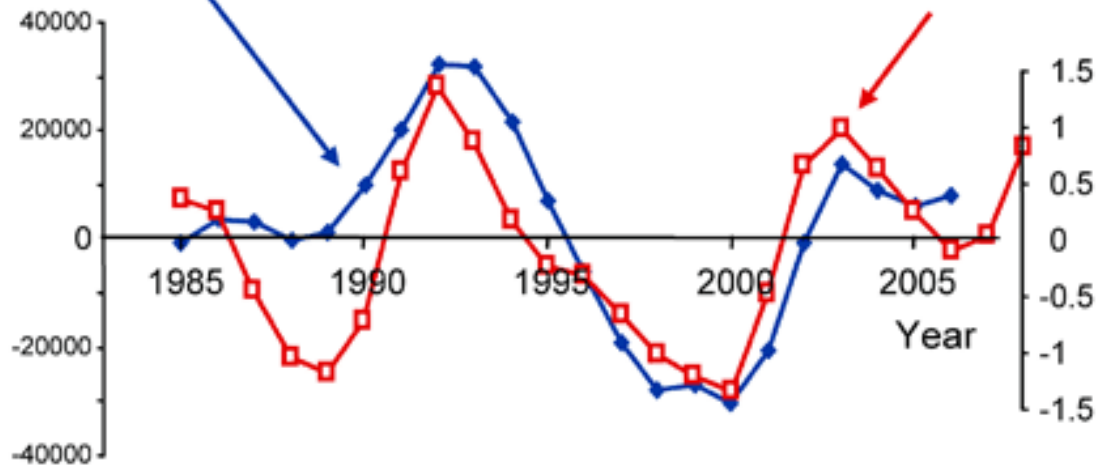
YES

Maybe

# Why you might want to go to grad school

Fluctuations in Grad Student Enrollment (Science & Engineering)

Fluctuations in the Unemployment Rate



Correlation  
Coefficient:

$$\rho = 0.75583$$

(that's pretty high)

grad school is for suckers! wall street here i come!



## Guess Who's Coming to Grad School?

Sources: NSF/Bureau of Labor Statistics. Fluctuations obtained by subtracting the mean regression line from the absolute values.

i always wanted to go to grad school...



# Why you might want to go to grad school (really)

- Passion for the material
  - And its **much** more fun than undergrad...
- Opportunity to work on **cutting edge problems**
- Interested in research, teaching, or starting a company (grad school required or big help)
- Better job prospects
  - MS → a good financial investment (esp in down mkt)
  - PhD → opportunity to work on cutting edge problems for rest of career

# What's cool about grad school?

- Problem space is **open**
  - You define problem and solution
  - Doesn't need to make money in 2 quarters
- Combine technologies and ideas **before** a market even exists (or may not exist)
  - E.g., technologies focused on social good
- Focused on the prototyping the idea; not the making a bulletproof product

# Grad student lifecycle

- Two years of classes, but
  - Usually no more than 2 classes @ time
  - Grades are different



# How to Apply to Graduate School

- Applications usually due in December
  - Allocate time to them early so it is not a rush
- Components
  - Letters of recommendation, GRE scores, GPA, statement of purpose, research/work experience
- The *single* most important thing are your letters of recommendation
  - Hmmm....

# Letters of Recommendation

- Supervisor from internship
  - Fine, but count less if not famous researcher
- Senior grad student
  - Can be worthwhile, in addition to faculty letter, if you worked with them
- Professor
  - Took their class (ok, but won't be personal)
  - *Worked with professor on research* (ideal!)

# How do you get research experience?

- Find out what's going on
  - Identify other undergraduates who have had research experience
  - Talk to your TA's, Rick Ord, anyone who will listen, etc...
- Talk to faculty who you're interested in
  - Schedule a time to talk to them
    - Explain what you're good at, what excited about
  - Ideally, they'll give you a well-defined project
  - If so, take it seriously; have to be motivated

# The Ideal Graduate School Applicant (and undergrad researcher)

- Hard working
- Creative
- Excellent communication skills
- Team player
- Intelligent
- Broad interests and knowledgeable in target area
  - Math, systems building, etc.
- Broad interests

# Where to Go

- Lots of great choices, apply widely
  - Important to have strength in your areas of interest (ask letter writers for advice)
  - Ok to switch area, but understand that grad school is focused on specialization
- Visit the school after admitted
  - School enthusiasm for your being there is important
- Who your adviser is matters much more than the “brand/ranking” of school

# Decision Process

- Make sure file is complete (remind letter writers)
- Lots of noise in the process
  - Many applications
  - No need in your particular area
- Well-timed note after application is complete matters
- Financial offer
  - Most PhD programs offer some form of support, MS funding opportunities too (more TA than RA)
    - Lots of noise here as well
  - Best indicator is to talk to current students in program

# CSE Stereotypes

- Type, type, program, program, CSE = Java
  - No: about solving problems creatively using computers
- Got to love computers as its all about 'em
  - No: just as astronomy is not merely about telescopes
- Got to sit behind a desk in a dark room all day
  - No: lots of work outdoors as with sensors
- Work by yourself; interact only with machines
  - No: very much a social process today, lots of schmoozing
- Jobs mostly writing games, no world impact
  - No: huge variety – robotics, bioinformatics, helping blind
- No intellectual challenge
  - No: very hard problems: vision, AI, P = NP problem

# Women in Computing

- Women were first programmers
- Grace Hopper and others were pioneers
- Two Turing Award winners (Fran Allen) and last (Barbara Liskov). Bios on Wikipedia
- UCSD has a Women in Computing Group with a Facebook page. They love to have UGs
- National association also has conferences and support groups <http://gracehopper.org>
- Great environment: lots of jobs that are part-time, telecommuting if you need them

# Unsure of your major?

- Choose CSE because
  - The content is intellectually rich, world-impacting, and is a substrate for work in many other fields
  - The people are cool, diverse, and fun
  - The jobs are high paying, plentiful, & enjoyable
- Choose CSE@UCSD because
  - The program has great courses, great opportunities, and is a bargain
  - The faculty are world-changers, cool, and really teach classes
  - Our students are already changing the world
- If you are unsure, try a CSE class at UCSD

# How we did CSE 91

- Various areas of CSE in each lecture and how they can solve the world's problems
- Each lecture was an appetizer:
  - Homework made you take a few bites
  - Lectures pointed to full meal (later courses)
  - It's an incredibly rich banquet
- You've taken the course, now stay the distance, and come **change the world** with us

# Computers have changed the world

- Libraries → Google
- Barnes and Noble → Amazon
- Credit Unions → Online banking
- Drop in at neighbor → Visit Facebook Page
- Soapbox in park → Blog to world
- Math with paper → Mathematica, Maple
- Hand drawing in art, movies → Graphics, Pixar
- Biology in wet lab only → DNA analysis
- ? → Its your turn

# You, too, can make the world better . .

