Welcome to CS88

David E. Culler

CS8 – Computational Structures in Data Science

http://inst.eecs.berkeley.edu/~cs88

Lecture 1

January 25, 2016

Goals today

- **Introduce you to the course**
- **Answer your questions**
- **CS Big Ideas**
  - **Algorithm**
  - **Data type**
  - **Representation**

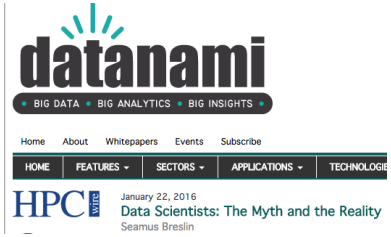
Data Science

Nearly every field of discovery is transitioning from "data poor" to "data rich"

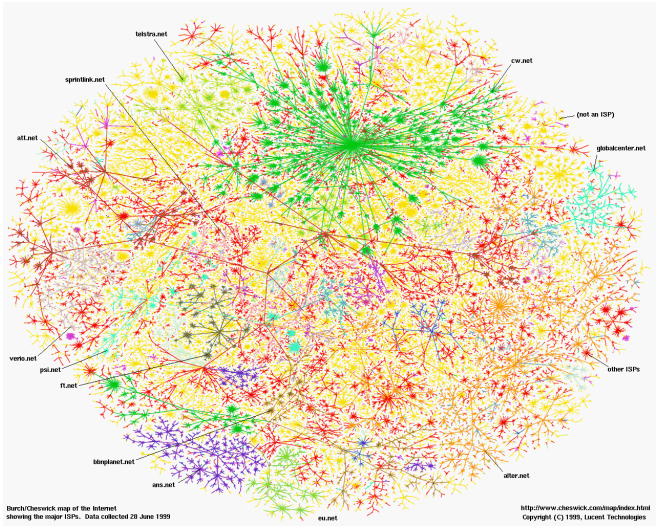
Data Science growing organically everywhere

Data Science

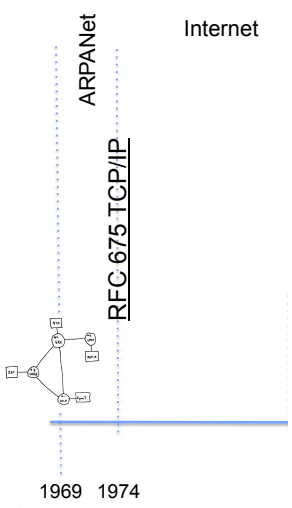
In the United States, it is reported that by 2018 there will be more than 490,000 data science positions available, but only 200,000 qualified people to fill the roles. The average size of a graduate class of data science students is 23 students. With approximately only 110 universities offering data science studies, the growing market will continue to pressure the supply in the US.

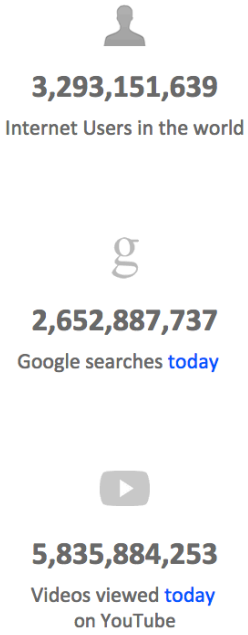


Greatest Artifact of Human Civilization …



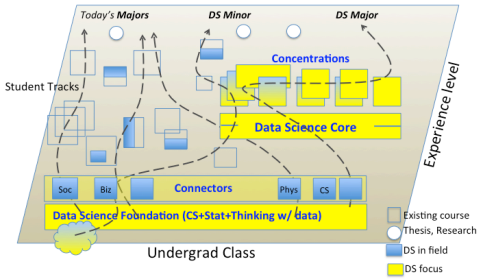
A Connecte





Data 8 – Foundations of Data Science

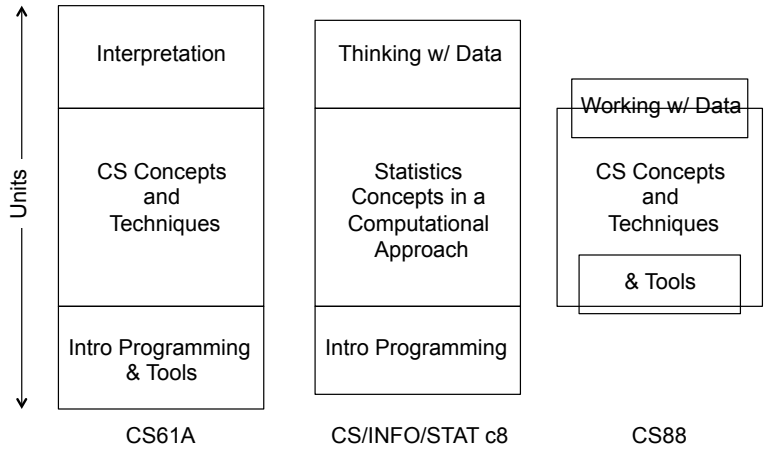
- **Computational Thinking + Inferential Thinking in the context of working with real world data**
- **Introduce you to several computational concepts in a simple data-centered setting**
  - **Authoring computational documents**
  - **Tables**
  - **Within Python3 and "SciPy"**



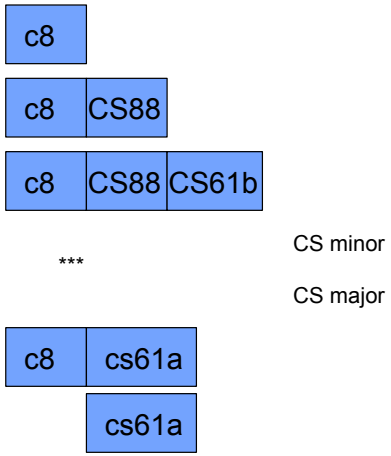
CS88 – Computational Structures in Data Science

- **Deeper understanding of the computing concepts introduced in c8**
  - **Hands-on experience => Foundational Concept**
  - **How would you create what you use in c8 ?**
- **Extend your understanding of the structure of computation**
  - **What is involved in interpreting the code you write ?**
  - **Deeper CS Concepts: Recursion, Objects, Classes, Higher-order Functions, Declarative programming, …**
  - **Managing complexity in creating larger software systems through composition**
- **Create complete (and fun) applications**
- **In a data-centric approach**

How does CS88 relate to CS61A ?

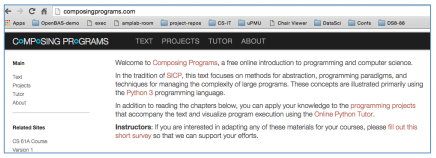


Opportunities for students



Course Structure

- 1 Lecture + 1 Lab/Discussion on Monday (!!!)
- Lecture introduces concepts (quickly)
- Lab provides concrete detail hands-on
- Homework (10) cements your understanding
  - Out Monday, Due Sunday
- Projects (3) put your understanding to work in building complete applications
  - Cuke
  - Maps
  - All about objects…



- Readings: composingprograms.com
  - Same as cs61a

Project 1: Cucumber (Agurk)

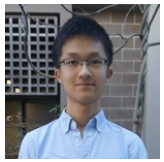
- Trick game: object = “don’t take last trick”
  - i.e., avoid getting in a pickle
- Deal 7 cards to each player
- Trick: must play >= largest played or lowest card
  - Suits don’t matter, points 2-14, Ace high
  - Last highest card wins
- “Winner” of last trick: score += high card
  - Players of equal card: subtract from score
- @21 points get cucumber
  - Reset to next highest score
  - Two cucumbers you lose
- We’ll build simulation and BOTS
  - Data Science => Strategy of play
  - BOTS will play-off, we’ll analyze strategies

CS88 Team

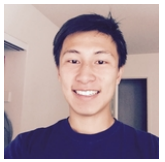
CS88 Team - uGSIs

Gunjan Baid
cunjan_baid@berkeley.edu

Nishant Desai
nishantdesai@berkeley.edu



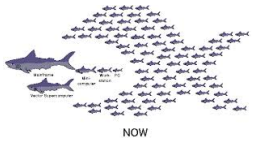
Ricky Liang
rickyliang@berkeley.edu



Garrett Tan
gtan@berkeley.edu

CS88 Team - me

- David Culler (culler@berkeley.edu)
  - 465 Soda Hall (amplab)
  - http://www.cs.berkeley.edu/~culler
  - Office hours: Tu 9-10, Fr 3-4 @ 511 Soda (hopefully)
  - Before/after class
- Build things
  - Cray Time Sharing System
  - OS386, OS286
  - Active Messages
  - Massive High Performance Clusters
  - TinyOS / Berkeley Motes, …
  - LoCal, BOSS, …



Course Culture

- Learning
- Community
- Respect
- Collaboration

Collaboration

Asking questions is highly encouraged
- Discuss all questions with each other (except exams)
- Submit lab assignments individually (graded on completeness)
  - If you come to lab, you can collaborate liberally
  - If you choose not to come to lab, you must work alone
- Submit homework individually and list collaborators
- Submit projects in pairs; find a partner in your lab

The Limits of collaboration
- Don't share solutions with each other (except project partners)
- Copying solutions will result in failing the course

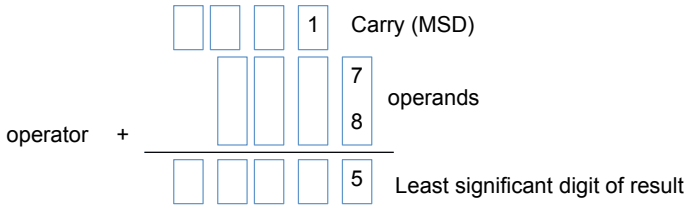
How do I get an account ?

- If you already have a cs8-* account, you are all set
- Otherwise: http://inst.eecs.berkeley.edu/webacct

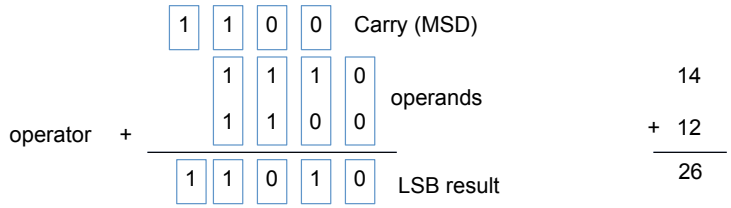
Algorithm

- An algorithm (pronounced AL-go-rith-um) is a procedure or formula for solving a problem.
- In mathematics and computer science, an algorithm is a self-contained step-by-step set of operations to be performed.
- An algorithm is an effective method that can be expressed within a finite amount of space and time and in a well-defined formal language for calculating a function.

Algorithms early in life



Algorithms early in life (in binary)



A Simple Algorithm in Class

- Count the number of students

More interesting one, …

- Betcha people in here share a birthday?

https://en.wikipedia.org/wiki/List_of_Presidents_of_the_United_States_by_date_of_birth

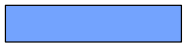
Presidents?

Data Type

- What's this?

Real (or ideal) world

42



Computer representation

Data Type

- Set of elements
  - with some internal representation
  - E.g. Integers, Floats, Booleans, Strings, …
- Set of operations on elements of the type
  - e.g. +, *, -, /, %, //, **
  - ==, <, >, <=, >=
- Properties
  - Commutative, Associative, … , Closure (???)
- Expressions are valid well-defined sets of operations on elements that produce a value of a type

Questions

- What's the difference between '==' and '=' ?

Lab and HW this week

- Lab will get you to where you have a *program development environment*
  - Even on your computer
- HW will give practice and explain subtleties of types, operators, and expressions
  - In a program development environment

Question of the week

- How many "things" can you represent with **N** bits