Announcements

HW6, HW7, and Lab 15 are up. All due May 5th.

- Your best 5 of 7 HWs and best 12 of 15 labs count.

Your job from now until the final: Study a little each day.

- Final exam is comprehensive.
- Struggling students: DO EVERY C LEVEL PROBLEM. All of them.
- Work with other people!
    - **Argue and discuss!**

¯_(ツ)_/¯

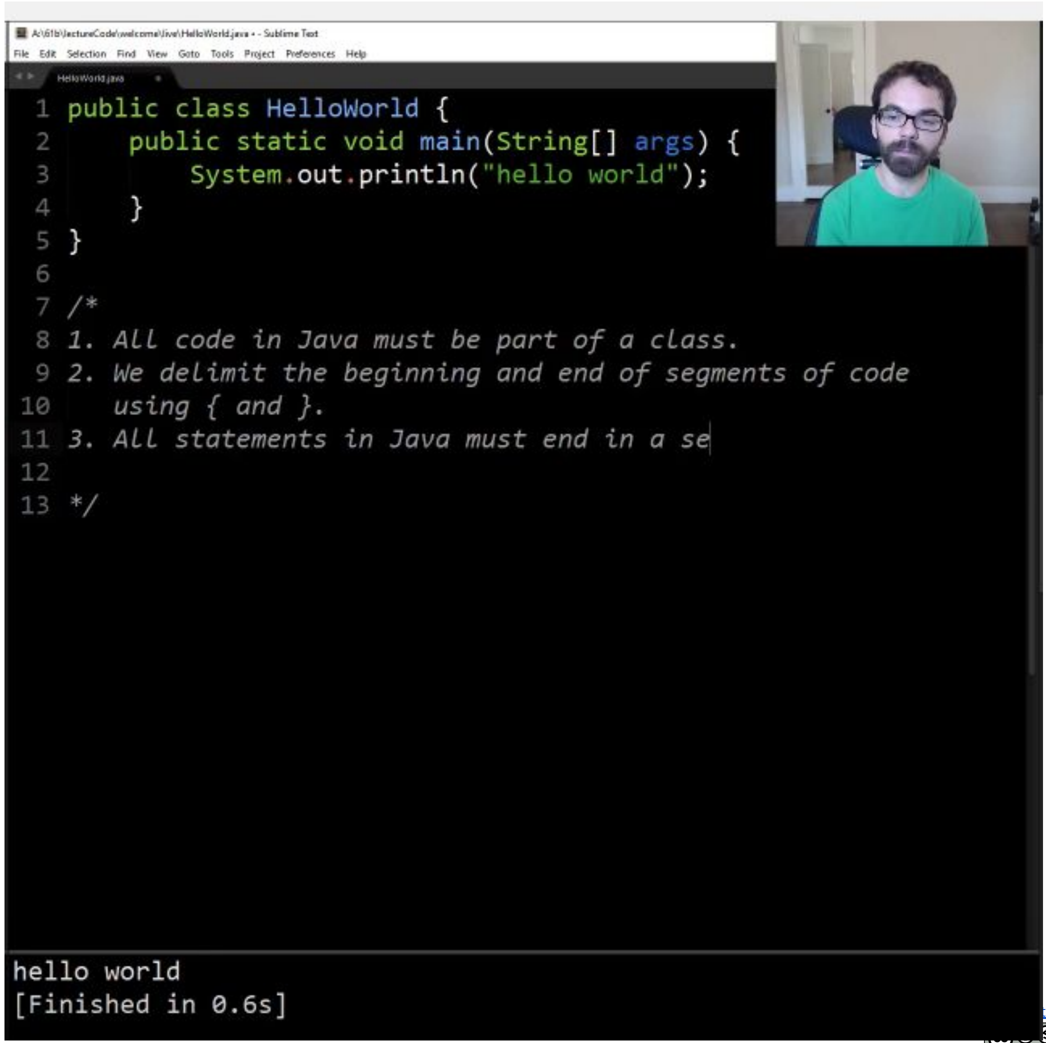
CS61B

Lecture 40: Wrapping Things Up

- What We've Done: 61B in 12 minutes or less.
- Moving Forwards.

Where We Started

Just 14 weeks ago:


A:\61b\lectureCode\welcome\live\HelloWorld.java • - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
HelloWorld.java

```java
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("hello world");
    }
}

/*
1. All code in Java must be part of a class.
2. We delimit the beginning and end of segments of code
   using { and }.
3. All statements in Java must end in a se

*/
```

```
hello world
[Finished in 0.6s]
```

What We've Learned about Programming Languages

- Object based programming: Organize around objects.
- Object oriented programming:
  - Interface Inheritance.
  - Implementation inheritance.
- Dynamic vs. static typing.
- Generic Programming, e.g. ArrayList<Integer>, etc.
- The model of memory as boxes containing bits.
- Bit representation of positive integers.
- Java.
- Some standard programming idioms/patterns:
  - Objects as function containers (e.g. Comparators, IntUnaryFunctions).
  - Default method specification in interfaces (Link).
  - Iterators and views (e.g. keySet).

Example: Programmer only needs to know List API, doesn't have to know that ArrayList secretly does array resizing.

Example: Array is a sequence of boxes. An array variable is a box containing address of sequences of boxes.

Important Data Structures in Java

Important data structure interfaces:

- java.util.Collection (and its subtypes).
  - With a special emphasis on Map (and its subtypes).
- Our own Collections (e.g. Map61B, Deque): Didn’t actually extend Collection.

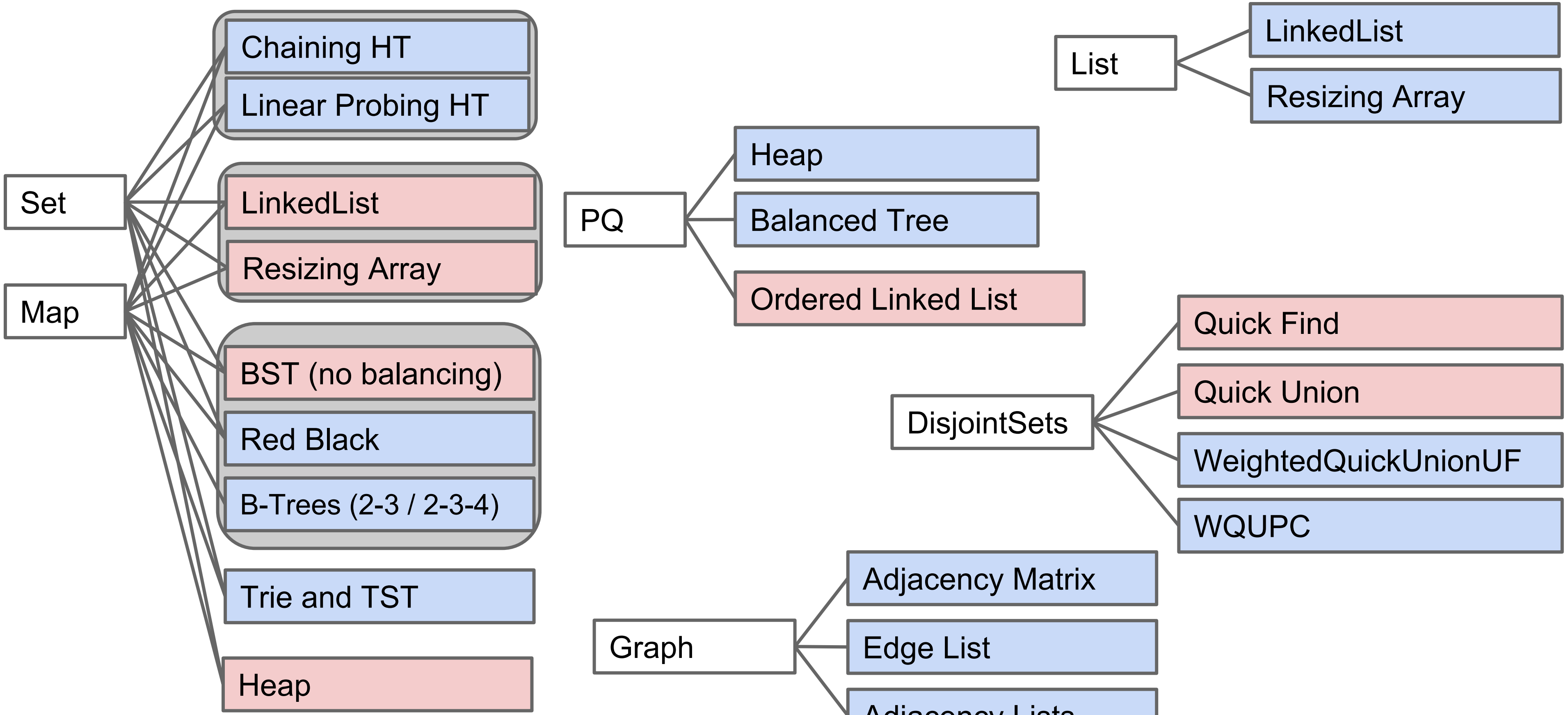
Concrete implementations of these abstract implementations:

- Examples: ArrayDeque implements Deque.

Mathematical Analysis of Data Structure/Algorithm Performance

- Asymptotic analysis.
- O(·), Ω(·), Θ(·), and tilde notation.
- Worst case vs. average case vs. best case.
  - Exemplar of usefulness of average case: Quicksort
- Determining the runtime of code through inspection (often requires deep thought).
- Amortized time. (Link)
  - Exemplar: ArrayLists are actually good at basic operations despite resizing. Amortized runtime is constant.

Some Examples of Implementations for ADTs



Arrays vs. Linked Data Structures

Array-Based Data Structures:

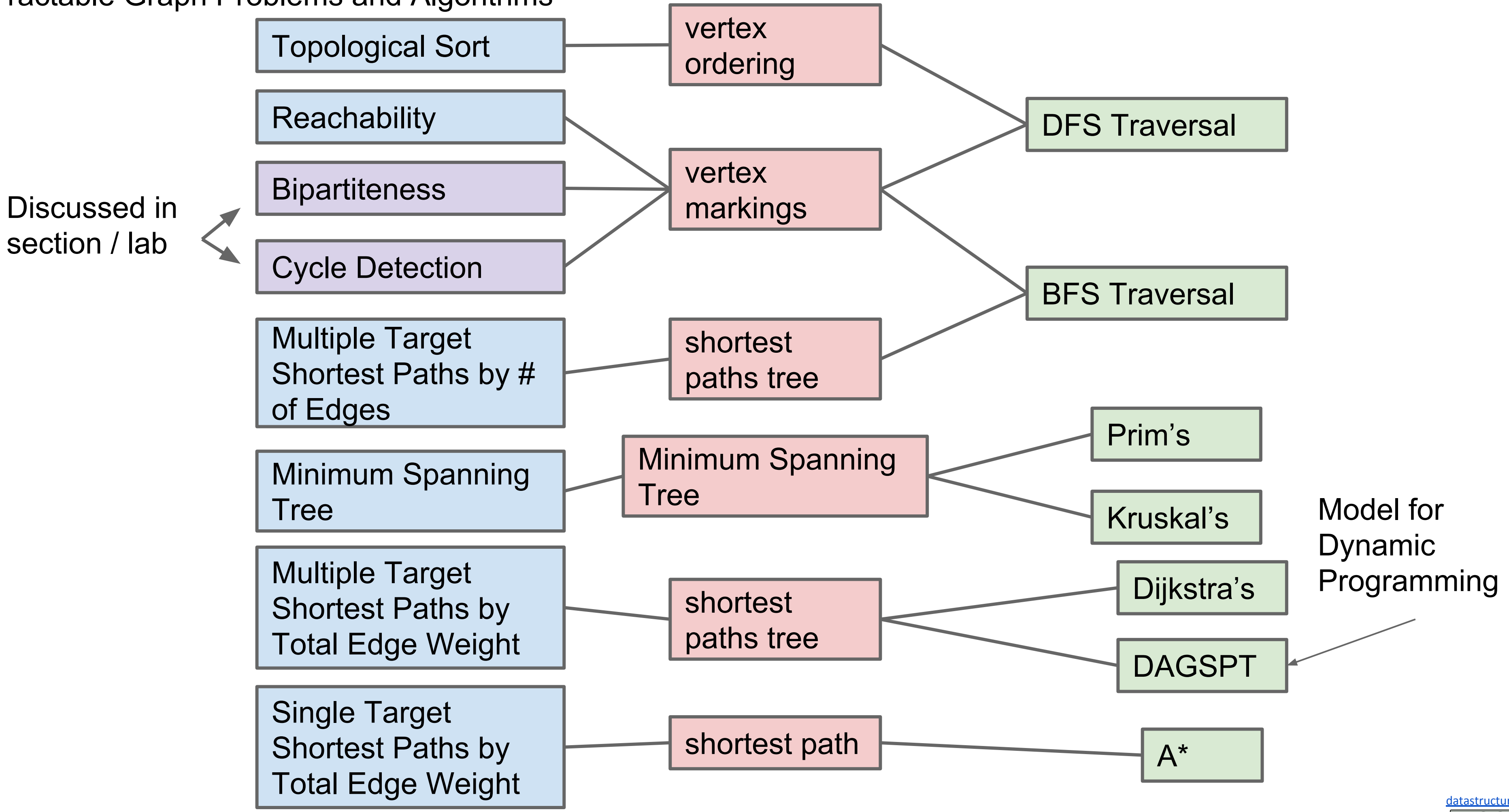
- ArrayLists and ArrayDeque
- HashSets, HashMaps, MyHashMap: Arrays of ‘buckets’
- ArrayHeap (tree represented as an array)

Linked Data Structures

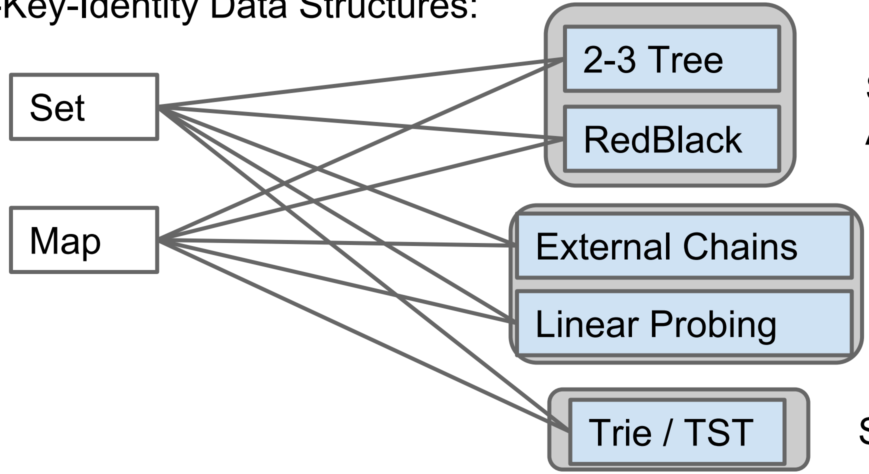
- Linked Lists
  - LinkedList, IntList, LinkedListDeque, SLList, DLList
- Trees: Hierarchical generalization of a linked list. Aim for bushiness.
  - TreeSet, TreeMap, BSTMap, Tries (trie links often stored as arrays)
- Graphs: Generalization of a tree (including many algorithms).

Tradeoffs of arrays vs. linked data structures.

Tractable Graph Problems and Algorithms



Search-By-Key-Identity Data Structures:

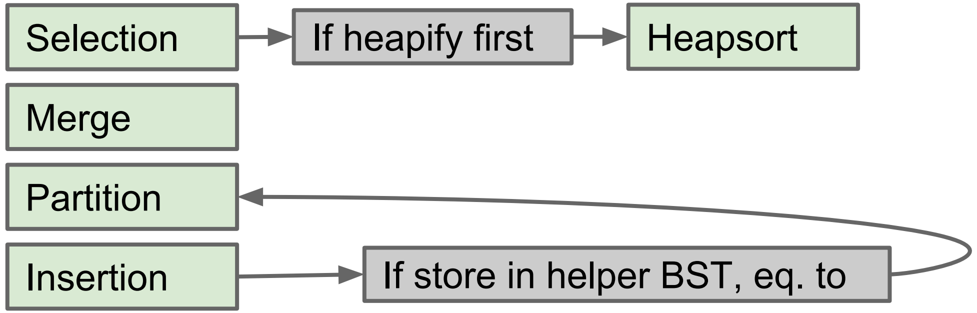


Searches using compareTo()
Analogous to **Comparison-Based**

Searches using hashCode() and equals()
Roughly Analogous to **Integer Sorting**

Searches by digit. Analogous to **Radix Sorting**

Comparison Based Sorting Algorithms: Ω(N log N) worst case

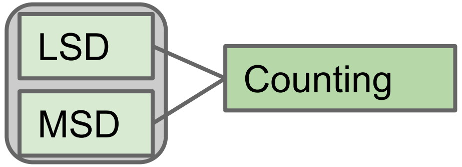


Small-Alphabet **(Integer) Sorting Algorithms**: Θ(N) worst case

Counting

Radix Sorting Algorithms: Θ(NL) worst case:

(require a sorting subroutine)



Fun/Weird Topics (This Week)

Compression:

- Huffman Coding, and selection of data structures for Huffman Coding.
- Other approaches: LZW and Run Length Encoding (extra slides).
- Observation: optimal compression of a stream of bits would provide a useful model of that stream of bits (e.g. hugPlant.bmp -> hugPlant.java).

Impossible and Intractable Problems:

- Impossible: Algorithms that finds best compression for any input.
- Intractable problems: 3SAT, Independent Set, NP-Completeness.
    - Does P = NP? Dramatic implications of a yes answer.

The Practice of Programming

- Java syntax and idioms.
- JUnit testing (and its more extreme form: Test-driven development).
- Mining the web for code.
- Debugging:
  - Identify the simplest case affected by the bug.
  - Hunt it down, giving it no place to hide.
  - With the right methodology, can find bugs even when finding bug through manual code inspection is impossible (see Horrible Steve in lab3).
- Real tools: IntelliJ, git, command line, Maven
- Data structure selection (and API Design)
  - Drive the performance and implementation of your entire program.
- Working with complex APIs, specifications: Project 2 and Project 3
  - Project 3 also involved interacting with an existing code base.

Questions About Anything?

- This baby is named Zela. We made that name up, because I am weird.
- Zela’s favorite data structure: milk.
- What conditioner do I use? I use Tresemme Wild and Wooly for Curly Boys.
- How did I get into Princeton? I did not, I was a bad high school student who once changed my failing grade in Algebra II and my parents busted me. I went to a state school (UT Austin). I used a pencil. It didn’t work.
- 5’4” 5 * 12 + 4 = 64. 3rd percentile.
- Zerg player because Zerg is good.
- Favorite song: Harry Hosono’s work is really just very nice.
- Favorite video game: Faster Than Light, Brogue, Kid Icarus, Mega Man series from NES. Actraiser for SNES.
- Funniest thing at UT Austin. Snowman was the thing was I was most proud of.
- Favorite CS course at CAL I don’t teach: Functional programming (61A). The 164 I have in mind.

Questions About Anything?

- Why does Matt Owen talk so loudly? He is excited (he says quietly).
- When i was a kid, i had no idea what i wanted to be… I thought veterinarian might be cool and picked the name “hug your pet”, true story, as my office. When Iw as 3, I said I wanted to work at NASA with my dad when I turned 33 (because then we would both be 33).
- Favorite avenger: hulk, he chill;.
- Favorite subreddit: yesyesyesyesno, nonononoyes
- Classes in the future: ds100 in the fall, 195 in the fall and maybe spring, and 61b in the spring, and who knows beyond that. They want me to teach 186 but i think it soudns boring -- joe is awesome and he will convince it is cool someday.
- Jackson as a TA? He seems very charming, but less self confident htan I ould expect given the charm.

Questions About Anything?

- Favorite megaman weapon: Summoning a dog is cool.
- How often do you shave? Every 5 or 6 days, because it’s easy to hide becasue I am small.
- Future of CS education: Massive classes that leverage the crowd in cool ways and intelligent tutoring systems that are better.
- Picture of me wihtout beard: post on piazza.
- 61C: Probably not soon, but Dan Garcia is awesome.
- Why did I come to Berkeley in grad school? People at other grad schools I was admitted told me to come to berkeley instead. Teaching: I feel like what I do matters here way more than at Princeton, where 89% of sutdenst are already set for life just being there.
- Why education? I think it plays to my strengths and not so many of my weakeness affect it.

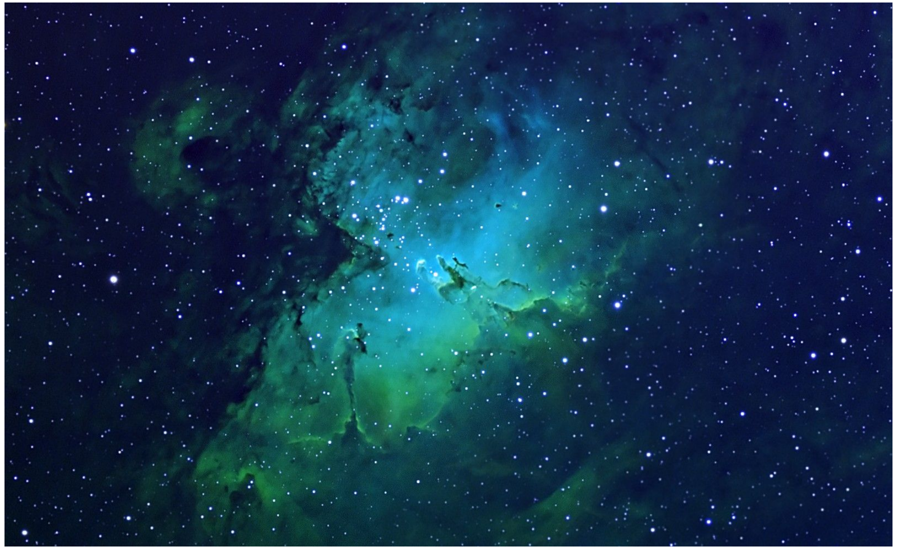
Questions About Anything?

- Favorite professor: John DeNero.

Moving Forwards

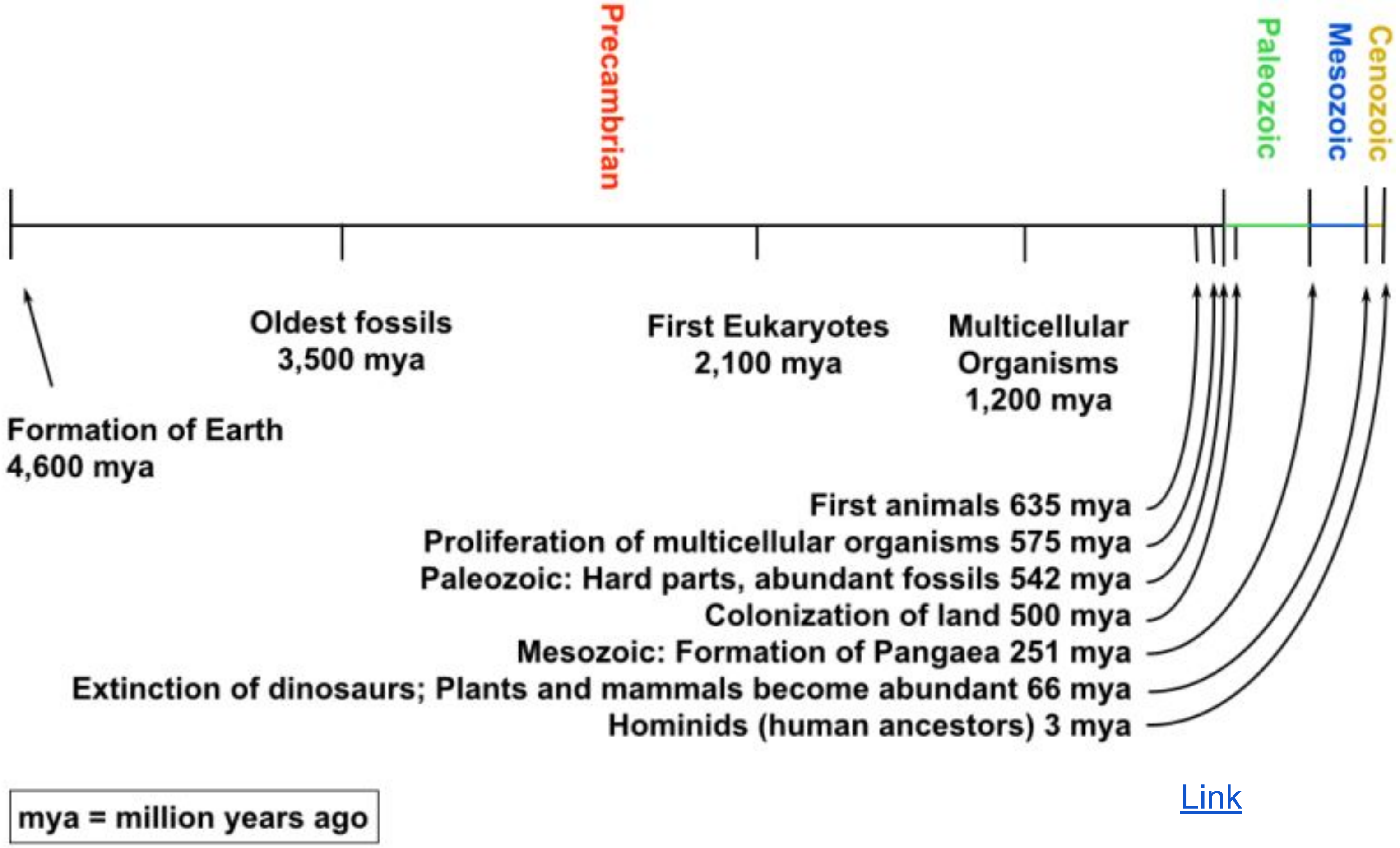
The Long View

Leaving a bunch of hydrogen lying around in space seems to result in rather complex outcomes.



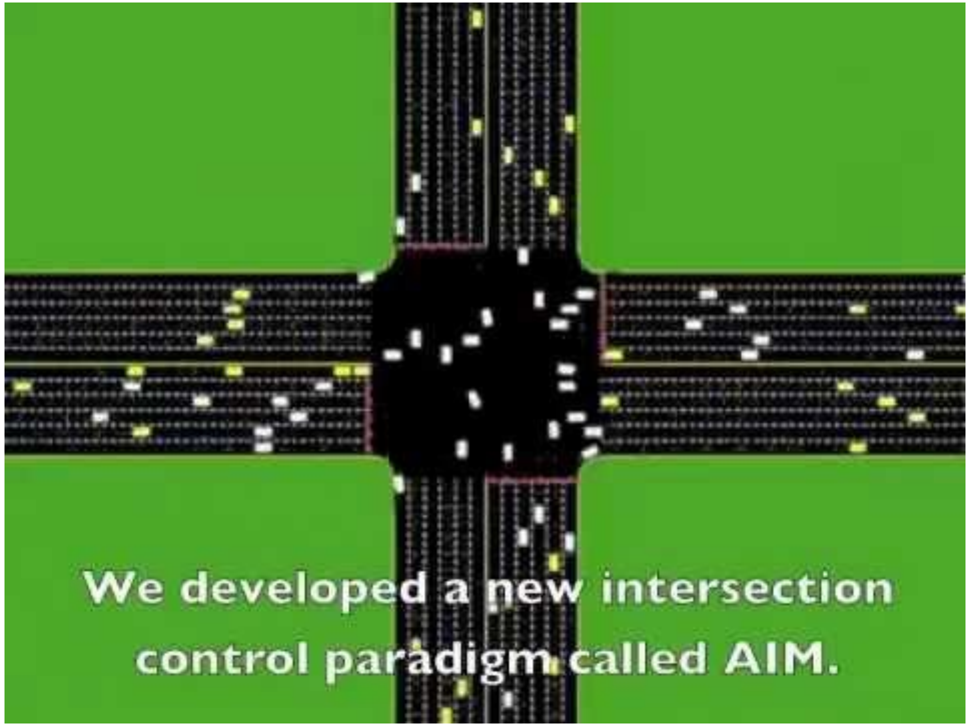
Link

The Long View: Accelerating Progress

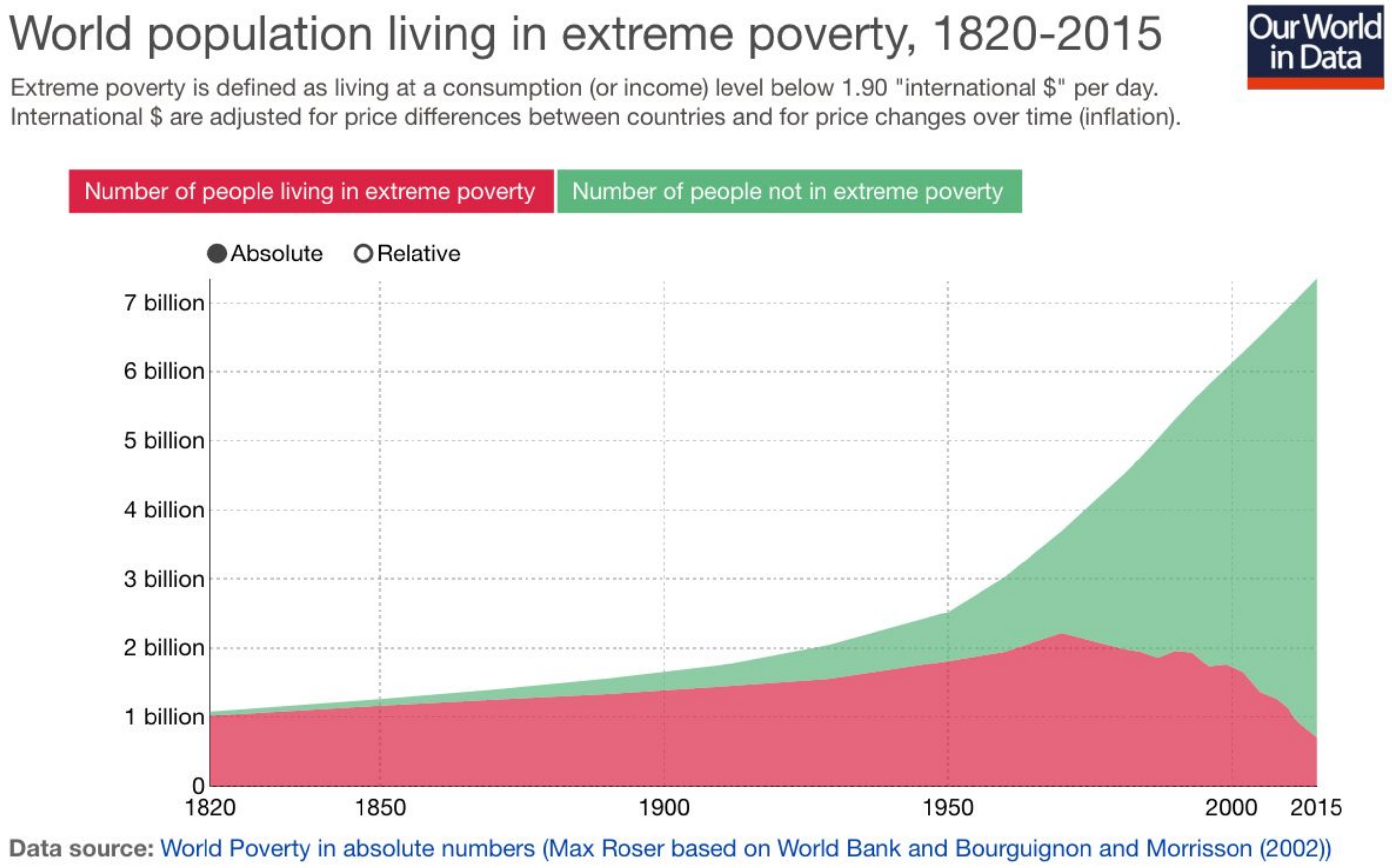

Precambrian
Paleozoic
Mesozoic
Cenozoic
Formation of Earth 4,600 mya
Oldest fossils 3,500 mya
First Eukaryotes 2,100 mya
Multicellular Organisms 1,200 mya
First animals 635 mya
Proliferation of multicellular organisms 575 mya
Paleozoic: Hard parts, abundant fossils 542 mya
Colonization of land 500 mya
Mesozoic: Formation of Pangaea 251 mya
Extinction of dinosaurs; Plants and mammals become abundant 66 mya
Hominids (human ancestors) 3 mya
mya = million years ago

Link

Computer Science is Fundamentally Changing Society


We developed a new intersection control paradigm called AIM.

The Good News: Global Poverty Has Declined Massively



The Bad News

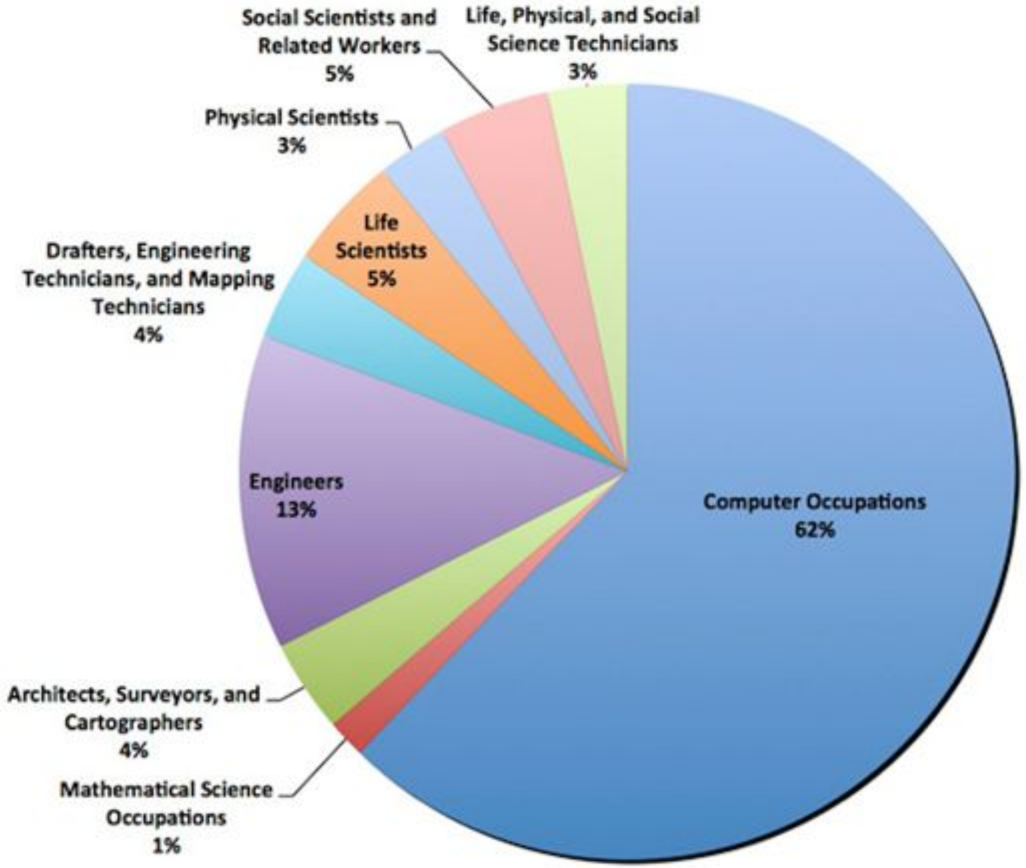
People are still mostly poor.

- Median worldwide income: ~$10,000/yr.
- Global 1% makes ~$64,000/yr.

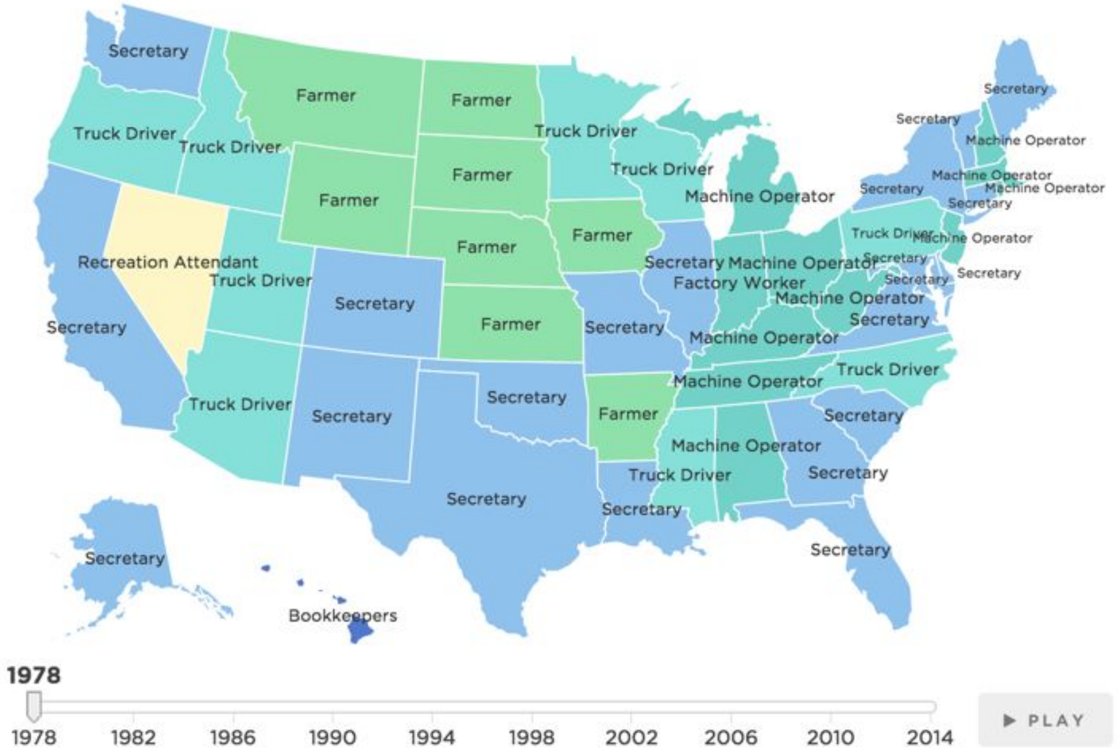
The United States (and the western world) seems to be growing unequal.

- Automation may play a large role in this process in the years to come.

Contribution to total growth in science and engineering occupations, 2010-2020 (Bureau of Labor Statistics)



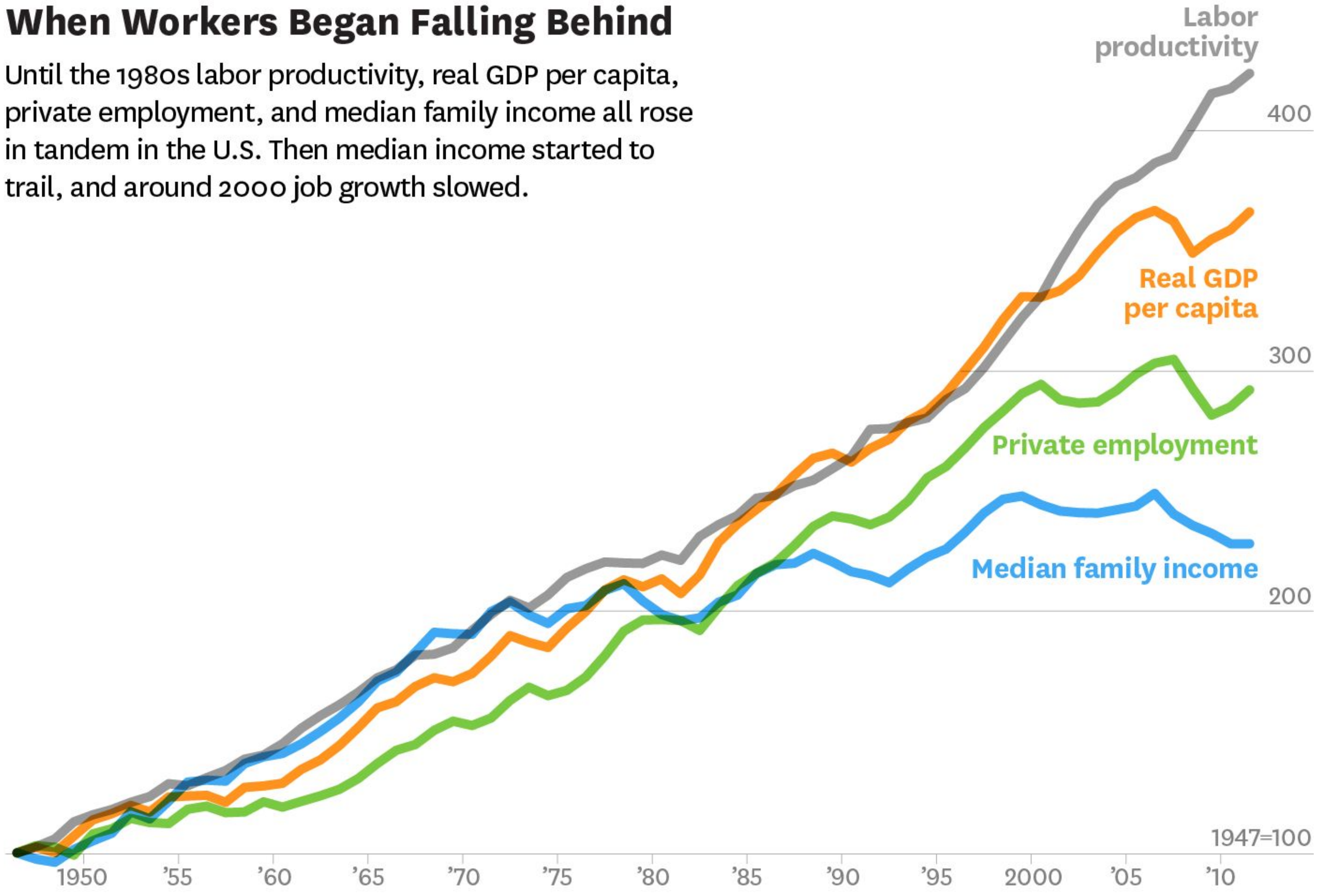
How is the Labor Pool Changing? (Statewide)



http://www.npr.org/sections/money/2015/02/05/382664837/map-the-most-common-job-in-every-state

When Workers Began Falling Behind

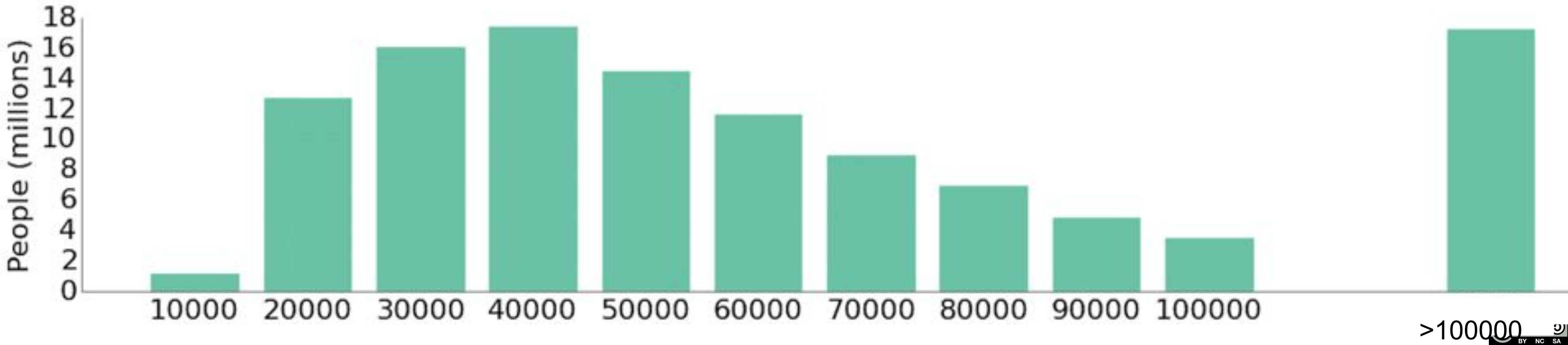
Until the 1980s labor productivity, real GDP per capita, private employment, and median family income all rose in tandem in the U.S. Then median income started to trail, and around 2000 job growth slowed.



SOURCE FEDERAL RESERVE BANK OF ST. LOUIS; ERIK BRYNJOLFSSON AND ANDREW MCAFEE
FROM "THE GREAT DECOUPLING," JUNE 2015

Income Inequality in the US

- Distribution of incomes of the ~111,123,000 people who worked full time, year round, data via Census Bureau.
- Highlights:
    - ~30+ million people made 30,000 or less [27%].
    - ~62+ million made 50,000 or less [55%].
    - ~17 million made 100,000 or more [15%].



Using Your Powers for Good

My request: Use your superpowers to improve the lives of humans.

- Not demanding that you work for low wages assisting the downtrodden (though that'd be great!).
- …. but keep in mind that your work will profoundly affect the world.

(Wanna talk about this stuff more? Take CS195!)

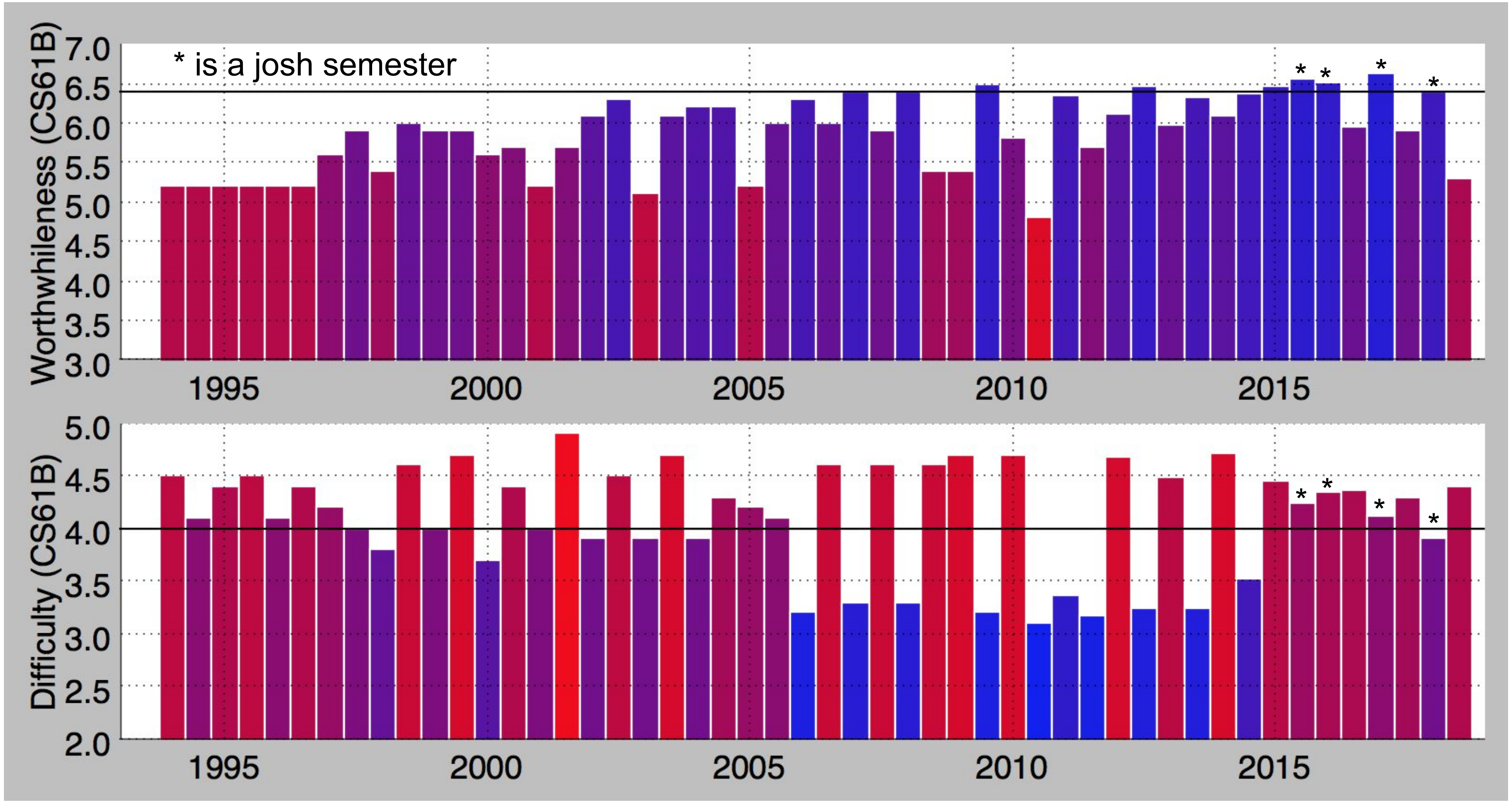
Course Reflections

Reflection on the Course: New for Spring 2018

- Online textbook and pre-recorded videos covering even more of the lectures.
- No terminal directions at all, everyone uses IntelliJ.
- Three types of discussion sections (LOST, exam prep and regular).
- Lots of extra enrichment exercises (proj0 and proj2 gold point videos, proj1 build your own autograder, proj3 driving directions).
- Project 2:
  - Made into an open ended project with required checkoff at the end.
  - Unlike that Piazza thread, previous projects did not involve a lot of tricky data structure selection, but were instead focuses on overall program organization and Class design.
  - Better team management (only ~3% unhappy teams vs. 10% last Spring).
  - In the past, project 2s were Gitlet, Text Editor, and Databases.
- Removed Quadtrees from Project 3.
- Vitamins.

Reflection on the Course: New for Spring 2018

On the Subject of Difficulty: 61B History (black line is Denero's 61A)



Things For Next Time

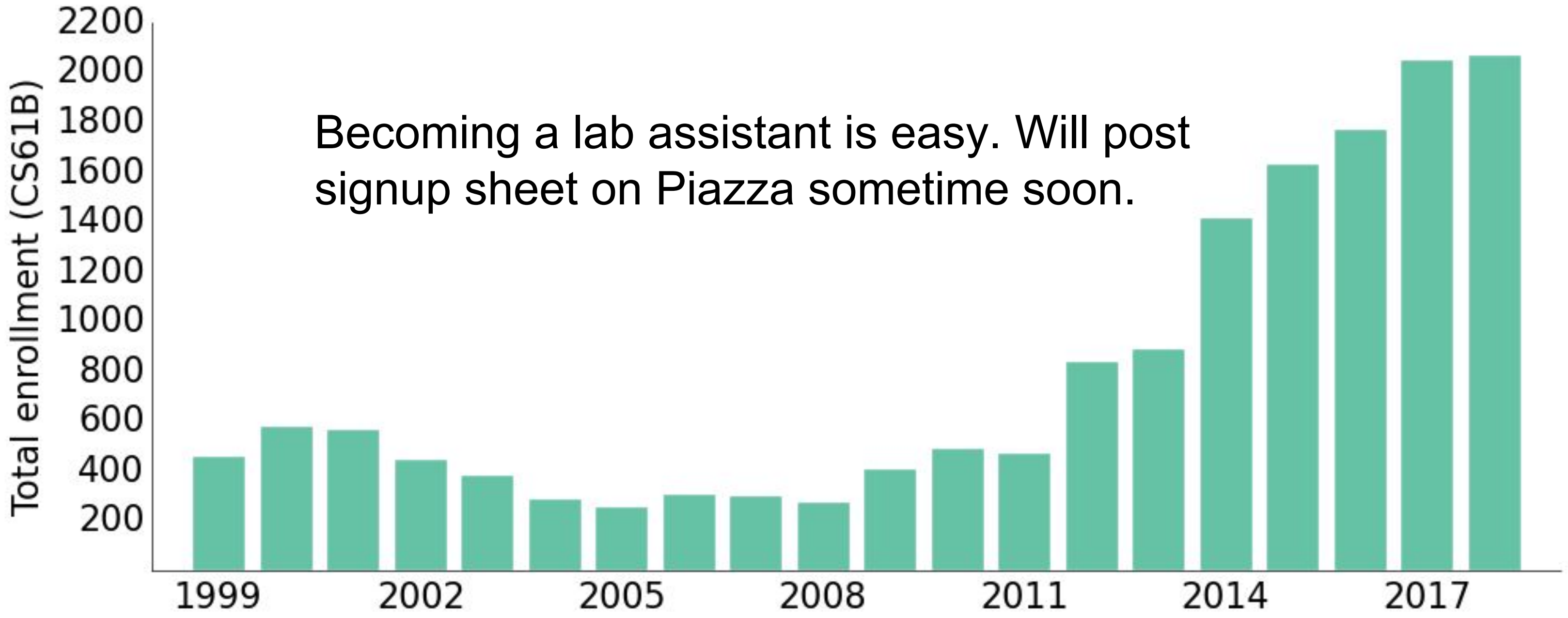
Possibilities:

- Semi-mandatory lab checkoffs.
- More flexible deadlines for HWs, labs, and projects.
- Cut out some of the boring Java syntax (e.g. protected keyword).
- Maybe add lambdas, function references, and/or streams (from Java 8).
- Tie project 2 more tightly to lecture content from weeks 4 - 7.
- Figure out how to teach students to debug more independently.
    - Would ideally like TA interactions to never look like "my code isn't working, help me."

Let me know what else you've missed on our official survey (coming out next week).

61B Needs You (Summer and Fall)

- Lab Assisting: Learn more, help others, get units, maybe become a GSI.
- Everyone is welcome, even if you're barely passing. (I mean it!)



Becoming a lab assistant is easy. Will post signup sheet on Piazza sometime soon.

Special thanks to the staff, without whom we would all be on fire.