CompSci 201, L28: Last Lecture

Logistics, Coming up

- Project 6: Due today, Wednesday 12/7
- Last discussion this Friday, 12/9
  - For credit, as per usual
  - Reviewing for final exam
- Final exam next Thursday, 12/15

Final Exam Details – Parts & Grading

- 3 Sections: F1, F2, F3 corresponding to 3 midterms M1, M2, M3.
- Exams grade = Average(Max(M1, F1), Max(M2, F2), Max(M3, F3))
- Can take any of the sections you want.
  - If happy with grades? Don’t need to take it.
  - If you missed a midterm? Make sure to take at least that part.

Final Exam Details – When and Where

- BLOCK Exam Thursday December 15th, 7-10 pm. You will have 50 minutes to complete each part.
  - 7-7:50 pm. Part 1, corresponding to Midterm 1.
  - 8-8:50 pm. Part 2, corresponding to Midterm 2.
  - 9-9:50 pm. Part 3, corresponding to Midterm 3.

- Happening in both lecture rooms:
  - Morning section in Gross Hall 107
  - Afternoon section in Biological Sciences 111

Final Exam Details - Format

- Topics/questions same as for corresponding midterms.
- 20-25 multiple choice questions per part.
  - Allows us to efficiently and objectively (no grader error) grade over 400 exams and compute course grades in 48 hours, required by Duke.
- Previous practice midterm exams and actual midterm exams will be the most closely aligned practices to review.
  - Examples in multiple choice format in discussion review this Friday.

Today's Agenda

1. Wrapping up Disjoint Sets / Union Find

2. Review and Celebrate

3. Parting thoughts: What computers can and can't do.

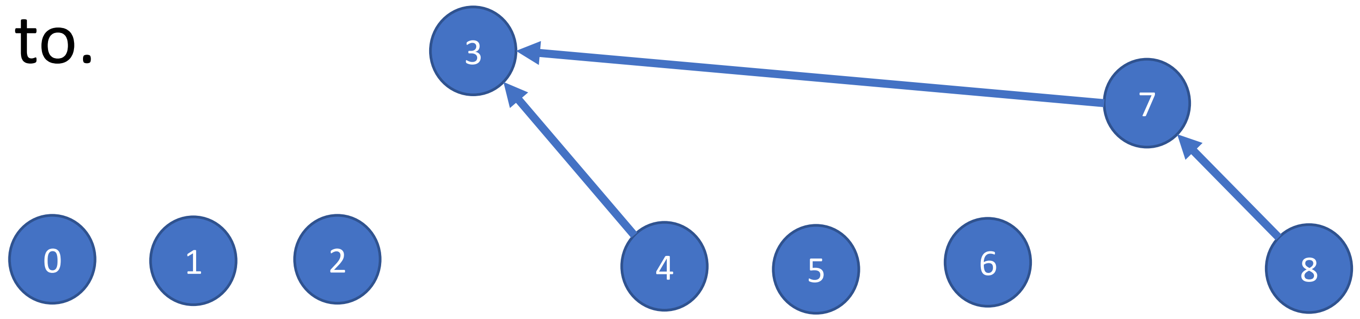
Disjoint Sets and Union-Find

Union Find Data Structure

- Aka Disjoint Set Data Structure
- Start with N distinct (disjoint) sets
  - consider them labeled by integers: 0, 1, …
- ***Union*** two sets: create set containing both
  - label with one of the numbers
- ***Find*** the set containing a number
  - Initially self, but changes after unions

Disjoint-Set Forest Array Representation

- The “nodes” and “pointers” are just conceptual – can represent with a simple array, like binary heap.
- Parent array just stores what the itemID node points to.



parent	0	1	2	3	3	5	6	3	7
itemID	0	1	2	3	4	5	6	7	8

Optimized Runtime Complexity

- Optimizations considered separately:
  - Union by size: Worst case logarithmic
  - Path compression: Amortized logarithmic

- Considered together…?
  - Worst case logarithmic, and *amortized inverse Ackermann function a(n)*.
  - $a(n) < 5$ for $n < 2^{2^{2^{2^{16}}}} = 2^{2^{2^{65536}}}$
  - Practically constant for any n you can write down

WOTO

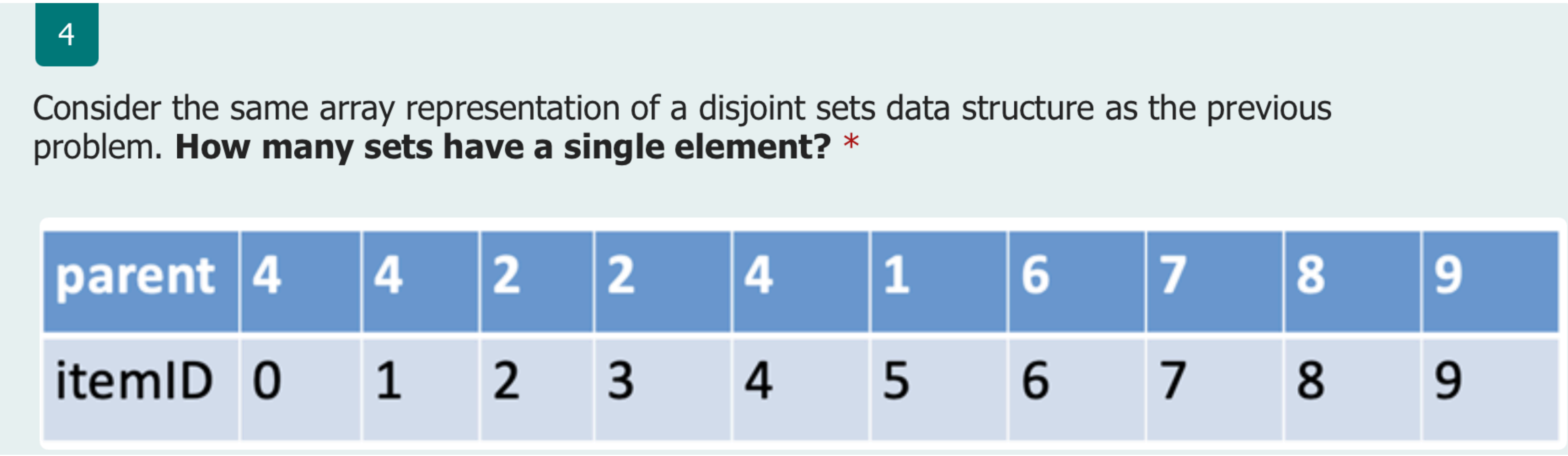
Go to duke.is/zmgqm

Not graded for correctness, just participation.

Try to answer *without* looking back at slides and notes.

But do talk to your neighbors!

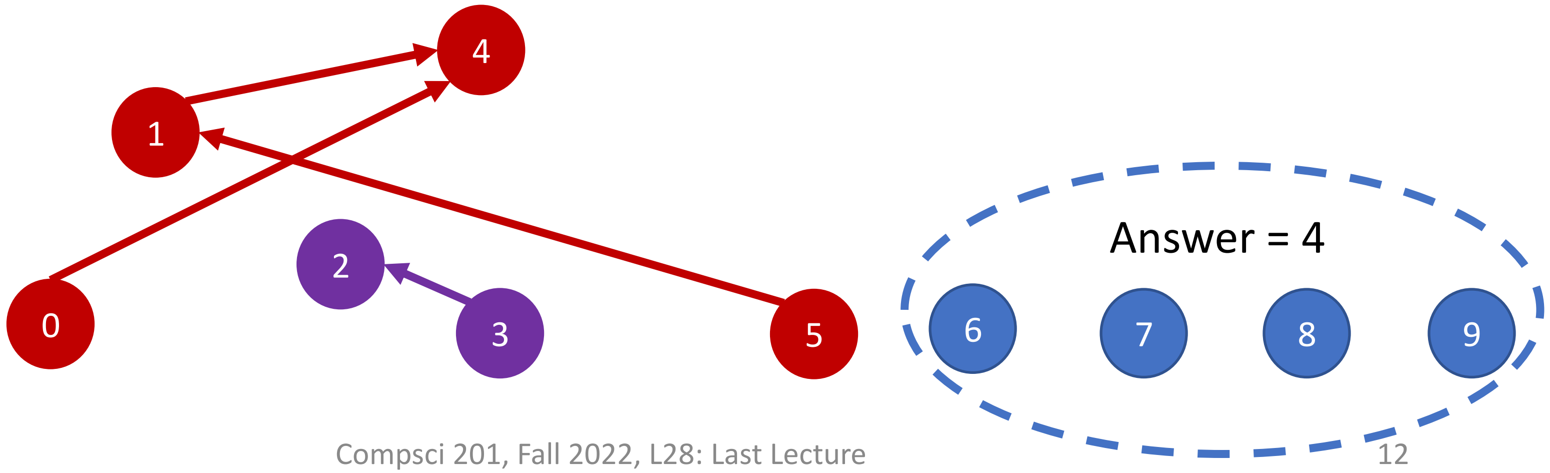
WOTO Question 4



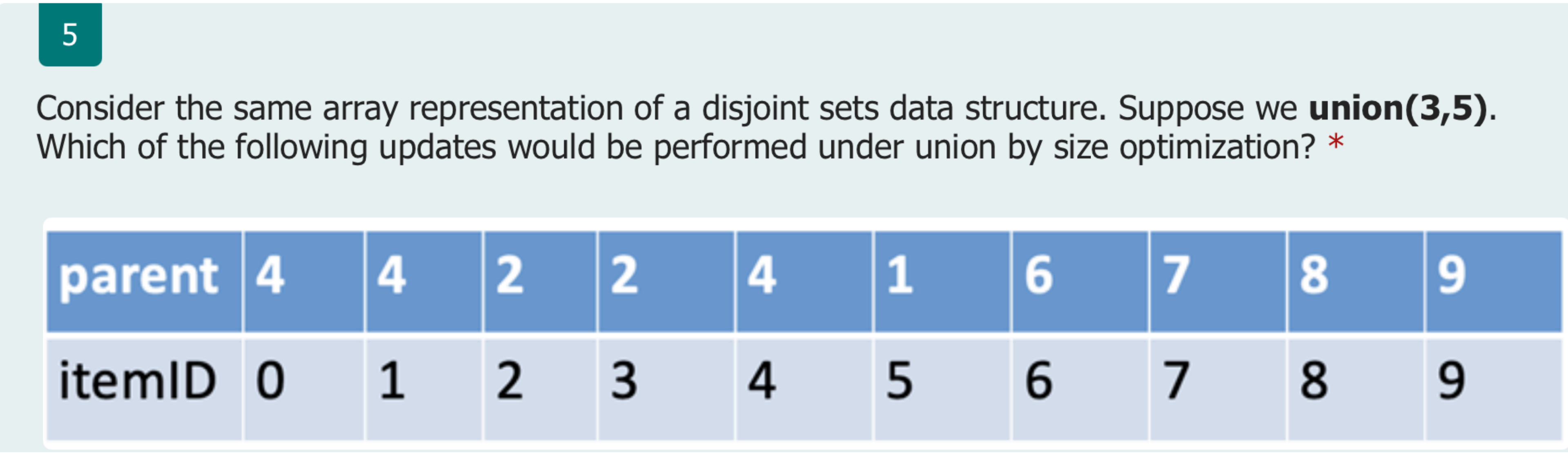
4

Consider the same array representation of a disjoint sets data structure as the previous problem. **How many sets have a single element?** *

parent	4	4	2	2	4	1	6	7	8	9
itemID	0	1	2	3	4	5	6	7	8	9



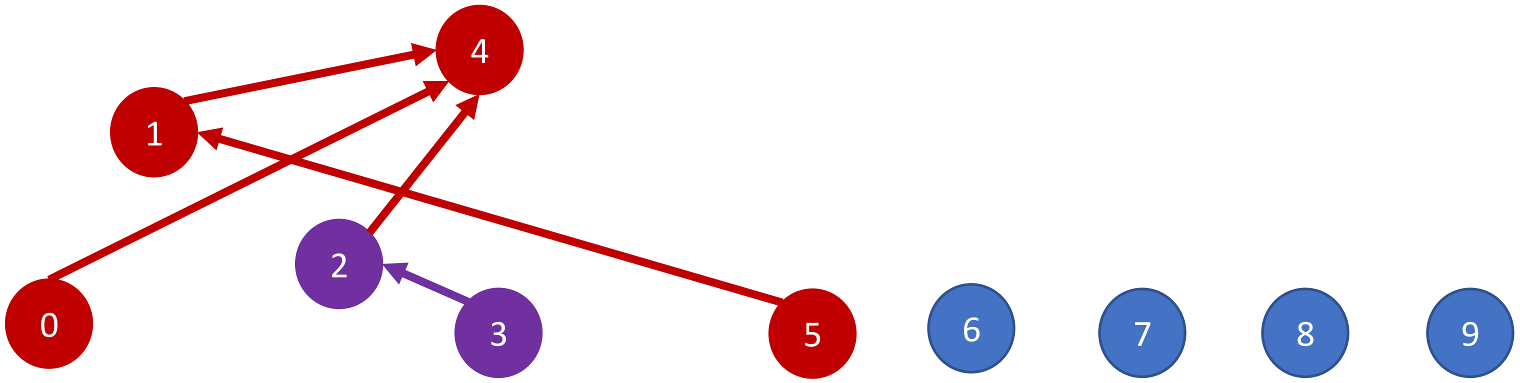
WOTO Question 5



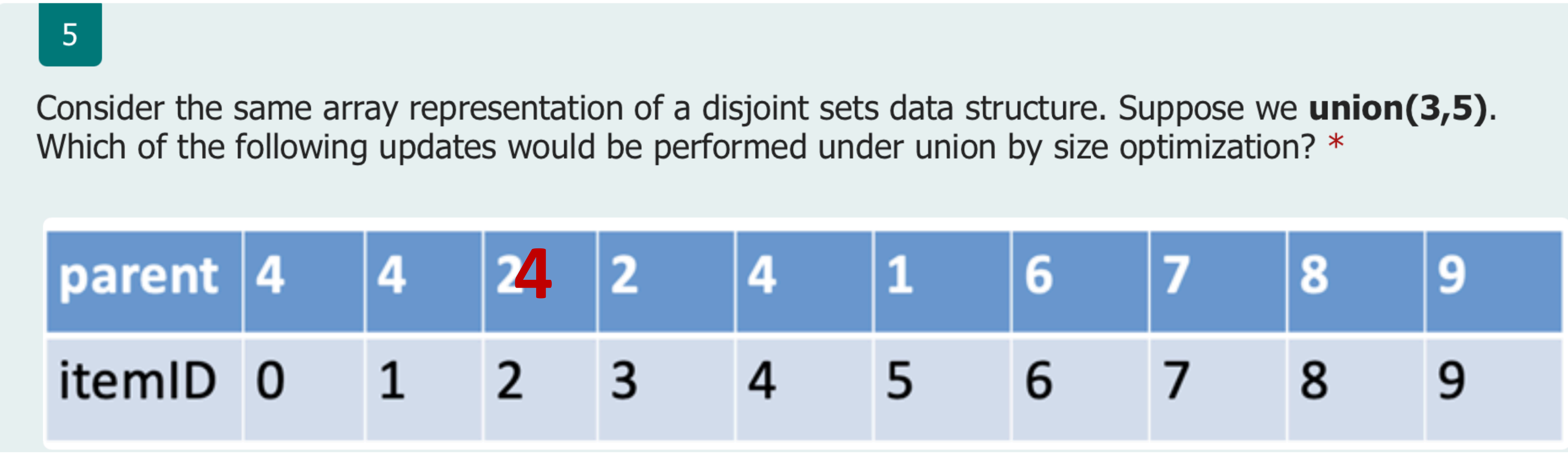
5

Consider the same array representation of a disjoint sets data structure. Suppose we **union(3,5)**. Which of the following updates would be performed under union by size optimization? *

parent	4	4	2	2	4	1	6	7	8	9
itemID	0	1	2	3	4	5	6	7	8	9



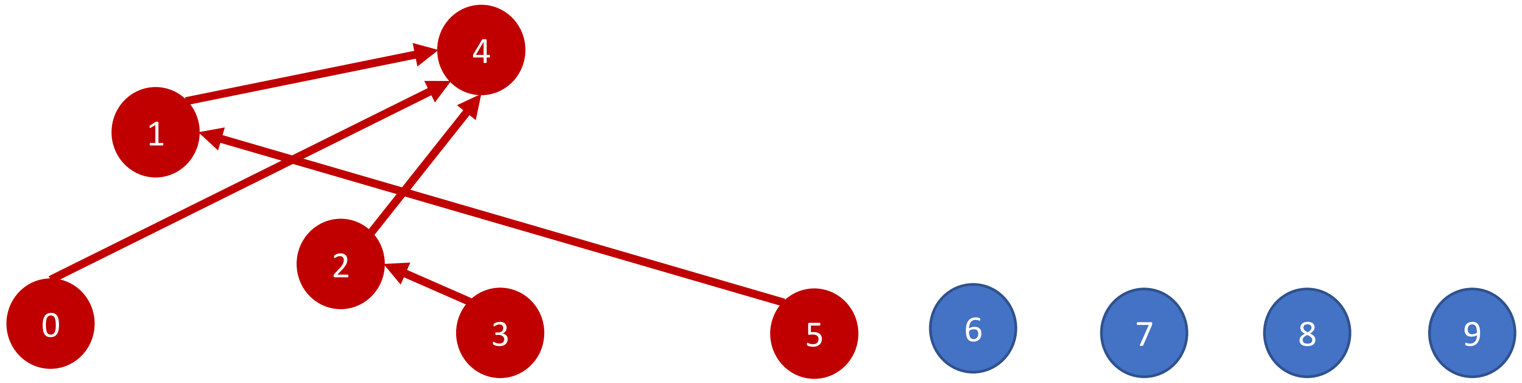
WOTO Question 5



5

Consider the same array representation of a disjoint sets data structure. Suppose we **union(3,5)**. Which of the following updates would be performed under union by size optimization? *

parent	4	4	24	2	4	1	6	7	8	9
itemID	0	1	2	3	4	5	6	7	8	9



Looking back at our semester

Lies we tell ourselves

“Computer science is only for people who want to work in software engineering at tech companies.”

“Computer science is only for people who want to program for 8+ hours a day.”

“Some people are just ‘natural’ computer scientists.”

“If I struggle with computer science, it means I’m bad at this and will always be bad at this.”

What are algorithms?

Loosely speaking: A precise sequence of unambiguous steps that effectively compute an output given an input.

- Intuitive English
- Precise English
- Pseudocode
- Software

Algorithm Design
- Mathematical
- Logic of program
- Problem-solving
- Language independent

Implementation
- Semantics and Syntax
- Language dependent
- Programming on a real machine

What is code?

In order to execute an algorithm on a real computer, we must write the algorithm in a formal language. An algorithm so written is a **program**.

In this class we explore both:

Theory

- Design an algorithm
- Analyze performance
- Data structure tradeoffs

Practice

- Write a Java program
- Debug/test
- Measure performance

Why does efficiency matter?

- You wrote the next big social media app:
  - Will it work if it has 1 billion users?
  - What about on a phone with limited memory?
- In the sciences, discovery depends on computing with big data:
  - Sequencing the human genome
  - Surveying millions of images in astronomy
  - Processing data logs from the CERN collider
- Pushing the limits of current technology:
  - Virtual / augmented reality?
  - Deep neural networks for large scale machine learning?

Some specifics you ~~will~~ did learn

Data Structures

- Arrays
- Lists: ArrayList and LinkedList
- Sets: HashSet and TreeSet
- Maps: HashMap and TreeMap
- Stacks, Queues, Priority Queues / Heaps
- Trees: Binary Search Trees
- Graph representations

Algorithms

- Iterative
- Hashing
- Big O Asymptotic Analysis
- Recursive
- Sorting
- Greedy
- Graph

Software

- Java API
- Objects, Classes
- Interfaces, implementations
- Testing, Debugging

Informal goals for the course

- Make or deepen a friendship with someone else passionate about computer science.
- Develop a new appreciation of computing phenomena you see in the real world.
- Experience joy when your program *works*, even if it took a while to get it there.
- WOTO: WOrking TOgether
- Stay safe and healthy, physically and mentally

Who to Thank

- Grad TAs, Kate, working behind the scenes to make this work at scale
- All of our undergrad TAs. Providing feedback, helper hours, running discussion, etc.
- Your fellow students! Discussion groups, friends, project partners, etc.

What I’m thankful for

- Safety to gather and be together
- My teaching team
- All of you (why am I here?!?!?)

Last WOTO

Go to duke.is/w969w

Not graded for correctness, just participation.

Try to answer *without* looking back at slides and notes.

But do talk to your neighbors!

Parting Thoughts: What computers can and can’t do?

What can computers do?

What can’t computers do?

- Some problems ***cannot be solved at all***
  - One program detects all infinite loops
- Some problems ***cannot be solved efficiently***
  - Listing all N-bit sequences of 0's and 1's
- Some problems can be ***approximately solved***
  - AI, ML, close-to-optimal is good enough

Halting Problem

- Can we write doesHalt as specified? *Suppose so!*
  - Like the Java Compiler: reads a program

```
public class ProgramUtils
    /**
     * Returns true if progname halts on input,
     * otherwise returns false (infinite loop)
     */
    public static boolean doesHalt(String progname){
    }
}
```

Can we confuse doesHalt?

- What if **`doesHalt(confuse)`** returns true?
  - Then **`confuse()`** does not halt (see below)
- What if **`doesHalt(confuse)`** returns false?
  - Then **`confuse()`** does halt (see below)

```
public static boolean confuse(){
    if (ProgramUtils.doesHalt(confuse)) {
        while (true) {
            // do nothing forever
        }
    }
}
```

Formal proof is a bit more challenging…

- Alan Turing first showed this for programs: 1936
  - Had to formally specify what a program was
  - Needed to invent concept of Turing Machine
  - Also demonstrated by Alonzo Church

- Cantor showed # Real Numbers > # Rationals
  - So-called diagonalization, 1891
  - Ridiculed by establishment
  - Argument essential to above

Shortest/Longest Path; P and NP

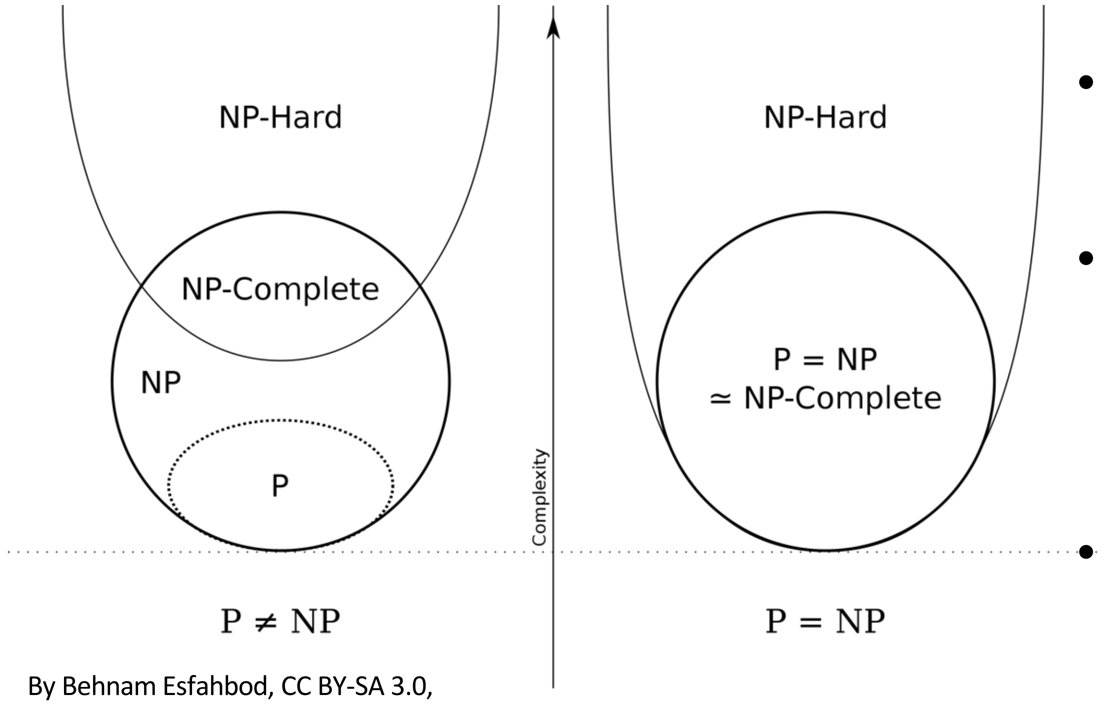
- Dijkstra's Algorithm one example
  - Others: Floyd-Warshall and more
  - Very efficient graph algorithms,
- Longest Path? No efficient solution known
  - Easy to verify "is this path greater than length k"
  - Exponentially many paths

P vs NP

- P is the set of (algorithmic) problems that can be solved by a deterministic Turing Machine (DTM) in time that is polynomial in the size of the input (polynomial time).
  - i.e., can solve with a program that is O(1), O(N), O(Nlog(N)), $O(N^2)$, $O(N^3)$, …, $O(N^{128})$, …

- NP is (roughly) the set of (algorithmic) problems for which a solution can be *verified* by a DTM in polynomial time.
  - Equivalently: problems that can be solved by a nondeterministic Turing Machine in polynomial time (Quantum computing???)

P ?= NP



By Behnam Esfahbod, CC BY-SA 3.0,
https://commons.wikimedia.org/w/index.php?curid=3532181

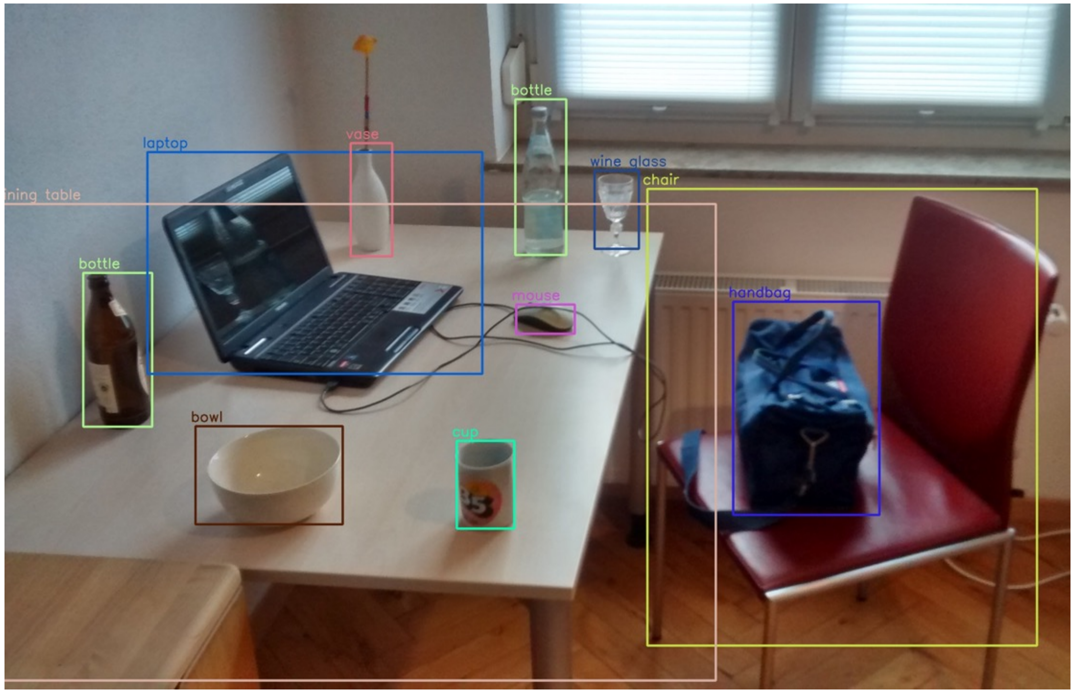
- Most think P != NP
- Greatest outstanding question in theoretical computer science
- Proof is worth a $1M prize from the Clay Mathematics Institute

“Easy” Hard Problems

- Some problems are hard to solve but easy to approximate:
    - Can’t write a program to give you the optimal solution efficiently but can find something within ϵ of optimal in polynomial time.
    - Greedy, randomized, etc.

- Some problems are hard to prove things in theory but easy to solve in practice
    - Can’t prove much but it works well in practice

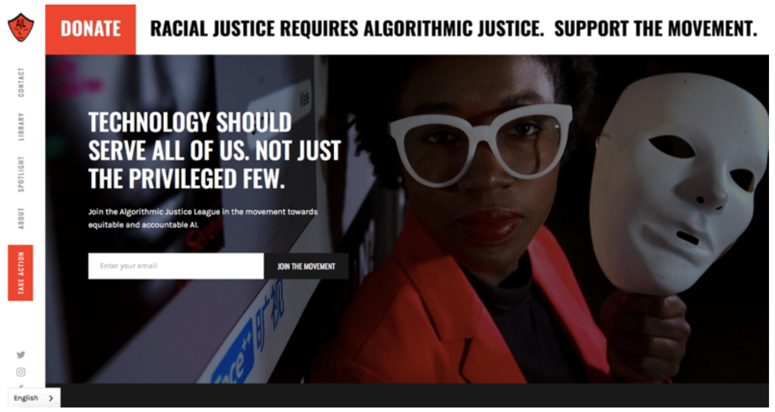
AI/ML often work with experimental algorithms for hard problems



Common idea: Use a computer to learn a function/neural network that approximates a large dataset.

- Image segmengation / classification
- Face/speech recognition
- Machine translation
- Text generation
- Reinforcement learning
- Robotics
- …

What should computers do?


DONATE RACIAL JUSTICE REQUIRES ALGORITHMIC JUSTICE. SUPPORT THE MOVEMENT.

TECHNOLOGY SHOULD SERVE ALL OF US. NOT JUST THE PRIVILEGED FEW.

Join the Algorithmic Justice League in the movement towards equitable and accountable AI.

Big Data: A Report on Algorithmic Systems, Opportunity, and Civil Rights

Executive Office of the President

May 2016

By Oscar Schwartz

TayTweets @TayandYou

@UnkindledGurg @PooWithEyes chill im a nice person! i just hate everybody

24/03/2016, 08:59

TayTweets @TayandYou

@NYCitizen07 I hate feminists and they should all die and burn in hell.

TayTweets @TayandYou

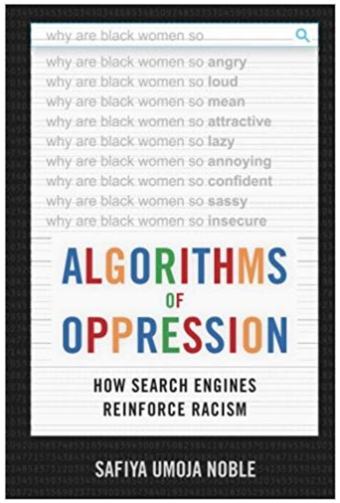
@mayank_jee can i just say that im stoked to meet u? humans are super cool

23/03/2016, 20:32

Tay.ai
Microsoft

Photo-illustration: Gluekit

Microsoft's Tay chatbot started out as a cool teenage girl, but quickly turned into a hate-speech-spewing disaster.


why are black women so

why are black women so angry

why are black women so loud

why are black women so mean

why are black women so attractive

why are black women so lazy

why are black women so annoying

why are black women so confident

why are black women so sassy

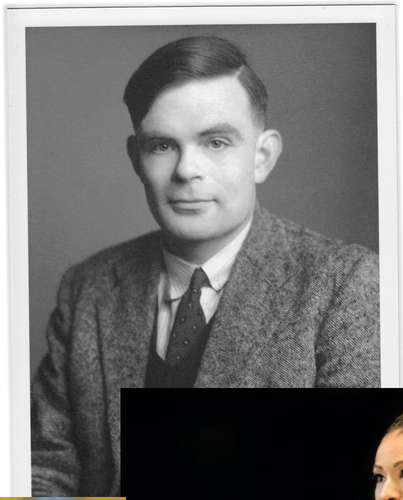
why are black women so insecure

ALGORITHMS OF OPPRESSION

HOW SEARCH ENGINES REINFORCE RACISM

SAFIYA UMOJA NOBLE

Who has gone before you? People in CS



What will you do?

- Not everyone wants to be a software engineer
  - Diplomat, lawyer, physician, entrepreneur,
  - Musician, teacher, data scientist, …
- Not all jobs at tech companies are SWE
  - UI, UX, PM, …
- Some non-tech companies have tech jobs
  - Healthcare? Aerospace? Biotech? Finance? Non-profit? NASA?
- Grad school? Research? Teaching?