DNA Helix Photomosaic from cover of *Nature*, 15 Feb 2001 (made by Eric Lander)

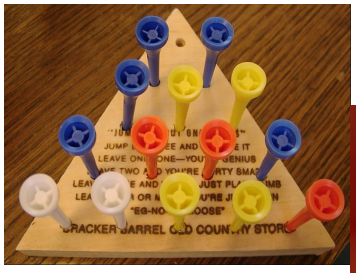
# Lecture 41: P = NP?

I will have extra office hours after class today (1-3pm). To be eligible to present Monday, your team must send an email with your URL before midnight tonight.

CS150: Computer Science
University of Virginia
Computer Science

David Evans
http://www.cs.virginia.edu/evans

## Pegboard Problem






## Pegboard Problem

- Input: a configuration of $n$ pegs on a cracker barrel style pegboard
- Output: if there is a sequence of jumps that leaves a single peg, output that sequence of jumps. Otherwise, output **false**.

How hard is the Pegboard Problem?



## Problems and Procedures

- To know a $O(f)$ bound for a problem, we need to find a $\Theta(f)$ *procedure* that solves it
  - The sorting **problem** is $O(n \log n)$ since we know a **procedure** that solves it in $\Theta(n \log n)$
- To know a $\Omega(f)$ bound for a **problem**, we need to prove that there is no procedure faster than $\Theta(f)$ that solves it
  - We proved sorting is $\Omega(n \log n)$ by reasoning about the number of decisions needed



## How much work is the Pegboard Problem?

- Upper bound: ($O$)
  $O(n!)$
  Try all possible permutations
- Lower bound: ($\Omega$)
  $\Omega(n)$
  Must at least look at every peg
- Tight bound: ($\Theta$)

No one knows!



## Complexity Class P "Tractable"

**Class P**: problems that can be solved in a polynomial ($O(n^k)$ for some constant $k$) number of steps by a deterministic TM.

Easy problems like sorting, making a photomosaic using duplicate tiles, simulating the universe are all in **P**.

## Complexity Class NP

**Class NP:** Problems that can be solved in a polynomial number of steps by a *nondeterministic* TM.

Omnipotent: If we could try all possible solutions at once, we could identify the solution in polynomial time.

Omniscient: If we had a magic guess-correctly procedure that makes every decision correctly, we could devise a procedure that solves the problem in polynomial time.



## NP Problems

- Can be solved by just trying all possible answers until we find one that is right
- Easy to quickly check if an answer is right
  - Checking an answer is in **P**
- The pegboard problem is in **NP**

  We can easily try $\sim n!$ different answers

  We can check if a guess is correct in $O(n)$ (check all $n$ jumps are legal)



## Is the Pegboard Problem in **P**?

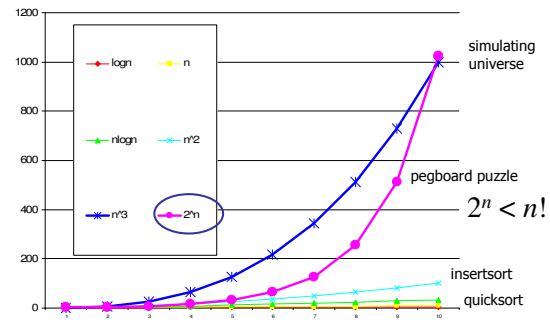
No one knows!

We can't find a $O(n^k)$ solution.

We can't prove one doesn't exist.

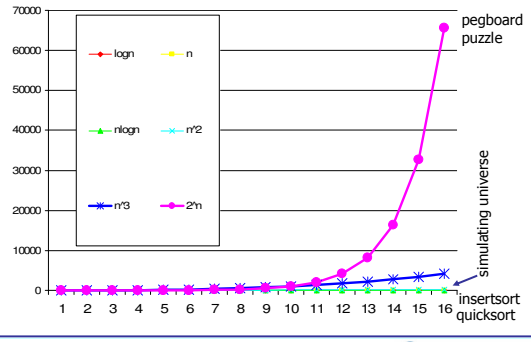


## Orders of Growth



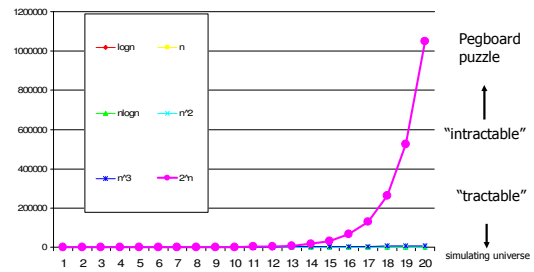



## Orders of Growth





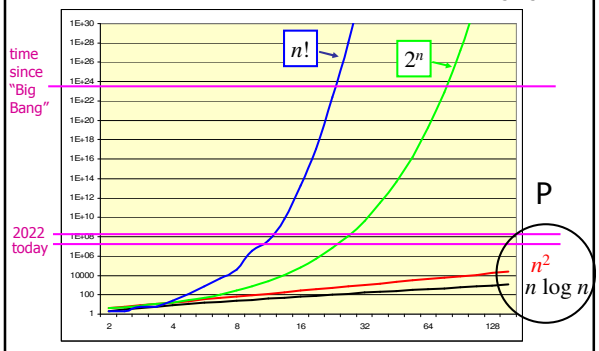

## Orders of Growth




*I do nothing that a man of unlimited funds, superb physical endurance, and maximum scientific knowledge could not do.*
– Batman (may be able to solve intractable problems, but computer scientists can only solve tractable ones for large $n$)

## Intractable Problems

log-log scale






## Moore's Law Doesn't Help

- If the fastest procedure to solve a problem is $\Theta(2^n)$ or worse, Moore's Law doesn't help much.
- Every ***doubling*** in computing power increases the solvable problem size by **1**.



## Complexity Classes

**Class P**: problems that can be solved in polynomial time by deterministic TM

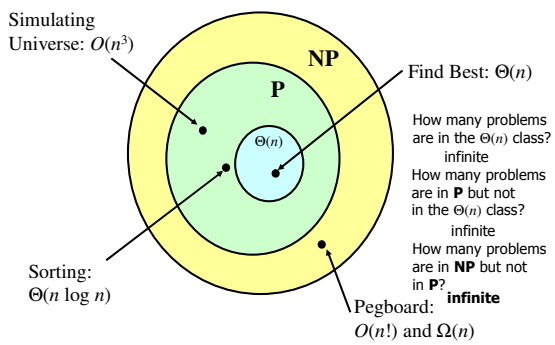
Easy problems like simulating the universe are all in **P**.

**Class NP**: problems that can be solved in polynomial time by a nondeterministic TM.

Includes all problems in **P** and some problems possibly outside **P** like the Pegboard puzzle.

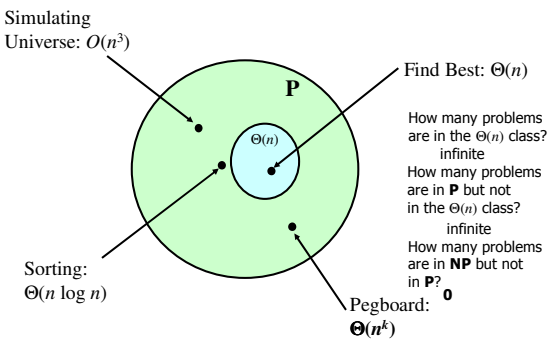


## Problem Classes if P ≠ NP:






## Problem Classes if P = NP:






## P = NP?

- Is P different from NP: is there a problem in NP that is not also in P
  - If there is one, there are infinitely many
- Is the "hardest" problem in NP also in P
  - If it is, then every problem in NP is also in P
- The most famous unsolved problem in computer science and math
  - Listed first on Millennium Prize Problems
  - $1M + an automatic A+ in this course

## NP-Complete

- **NP-Complete**: is the class of problems that are the *hardest* problems in **NP**
- Cook and Levin proved that 3SAT was NP-Complete (1971)
  - If 3SAT can be transformed into a different problem in polynomial time, than that problem must also be **NP-complete**.
  - Pegboard ⇔ 3SAT
- **Either all NP-complete problems are tractable (in P) or none of them are!**



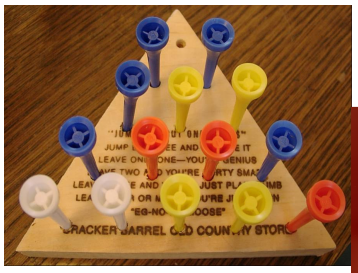
## NP-Complete Problems

- Easy way to solve by trying all possible guesses
- If given the "yes" answer, quick (in P) way to check if it is right
- If given the "no" answer, no quick way to check if it is right
  - No solution (can't tell there isn't one)
  - No way (can't tell there isn't one)

This part is hard to prove: requires showing you could use a solution to the problem to solve a known NP-Complete problem.



## Pegboard Problem






## Pegboard Problem

- Input: a configuration of $n$ pegs on a cracker barrel style pegboard
- Output: if there is a sequence of jumps that leaves a single peg, output that sequence of jumps. Otherwise, output **false**.

If given the sequence of jumps, easy ($O(n)$) to check it is correct. If not, hard to know if there is a solution.



## Most Important Science/Technology Races

1930-40s:Decryption Nazis vs. British
Winner: British
Reason: Bletchley Park had computers (and Alan Turing), Nazi's didn't

1940s: Atomic Bomb Nazis vs. US
Winner: US
Reason: Heisenberg miscalculated, US had better physicists, computers, resources

1960s: Moon Landing Soviet Union vs. US
Winner: US
Reason: Many, better computing was a big one

1990s-2001: Sequencing Human Genome



## Human Genome Race

Francis Collins (Director of public National Center for Human Genome Research) (Picture from UVa Graduation 2001)

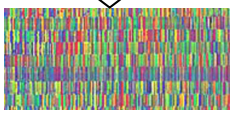
vs.

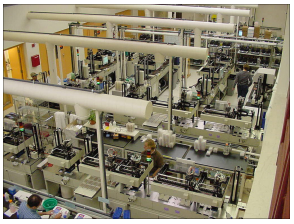
Craig Venter (President of Celera Genomics)

- UVa CLAS 1970
- Yale PhD
- Tenured Professor at U. Michigan

- San Mateo College
- Court-martialed
- Denied tenure at SUNY Buffalo

## Reading the Genome





Whitehead Institute, MIT



## Gene Reading Machines

- One read: about 700 base pairs
- But…don't know where they are on the chromosome

Read 3 TACCCGTGATCCA

Read 2 TCCAGAATAA

Read 1 ACCAGAATACC

Actual Genome AGGCATACCAGAATACCCGTGATCCAGAATAAGC



## Genome Assembly

Read 1 ACCAGAATACC

Read 2 TCCAGAATAA

Read 3 TACCCGTGATCCA

Input: Genome fragments (but without knowing where they are from)

Ouput: The full genome



## Genome Assembly

Read 1 ACCAGAATACC

Read 2 TCCAGAATAA

Read 3 TACCCGTGATCCA

Input: Genome fragments (but without knowing where they are from)

Ouput: The smallest genome sequence such that all the fragments are substrings.



## Common Superstring

Input: A set of $n$ substrings and a maximum length $k$.

Output: A string that contains all the substrings with total length $\leq k$, or no if no such string exists.

ACCAGAATACC

TCCAGAATAA

TACCCGTGATCCA

$n = 26$

ACCAGAATACC

TCCAGAATAA

TACCCGTGATCCA

ACCAGAATACCCGTGATCCAGAATAA



## Common Superstring

Input: A set of $n$ substrings and a maximum length $k$.

Output: A string that contains all the substrings with total length $\leq k$, or no if no such string exists.

ACCAGAATACC

TCCAGAATAA

TACCCGTGATCCA

$n = 25$

**Not possible**

## Common Superstring

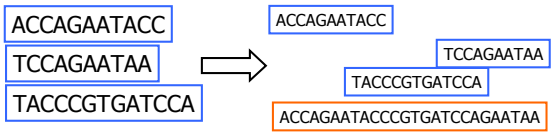
- In **NP:**
  - Easy to verify a "yes" solution: just check the letters match up, and count the superstring length
- In **NP-Complete:**
  - Similar to Pegboard Puzzle!
  - Could transform Common Superstring problem instance into Pegboard Puzzle instance!



## Shortest Common Superstring

Input: A set of $n$ substrings

Output: The shortest string that contains all the substrings.






## Shortest Common Superstring

Also is **NP-Complete**:

```
def scsuperstring (pieces):
  maxlen = sum of lengths of all pieces
  for k in range(1, maxlen):
    if (commonSuperstring (pieces, k)):
      return commonSuperstring (pieces, k)
```



## Human Genome

- 3 Billion base pairs
- 600-700 bases per read
- ~8X coverage required

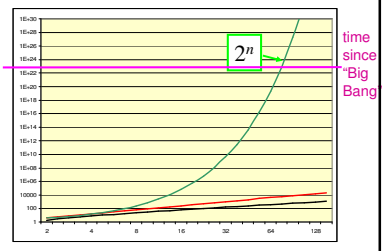
  > (/ (* 8 3000000000)) 650)
  36923076 12/13

- So, $n \approx$ 37 Million sequence fragments
- Celera used 27.2 Million reads (but could get more than 700 bases per read)



## Give up?

No way to solve an NP-Complete problem (best known solutions being $O(2^n)$ for $n \approx$ 20 Million)






## Approaches

- Human Genome Project (Collins)
  - **Change problem:** Start by producing a genome map (using biology, chemistry, etc) to have a framework for knowing where the fragments should go
- Celera Solution (Venter)
  - **Approximate:** we can't guarantee finding the shortest possible, but we can develop clever algorithms that get close most of the time in $O(n \log n)$

## Result: Draw?


President Clinton announces Human Genome Sequence essentially complete, June 26, 2000



## Charge

- Presentation qualification before midnight tonight