The Simpson's: $P = NP$?

Copyright © 1990, Matt Groening

1. Definition of $P$

$P$: Decision problems for which there is a polynomial algorithm.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Description</th>
<th>Algorithm</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>MULTIPLE</td>
<td>Is $x$ a multiple of $y$?</td>
<td>Grade school division</td>
<td>51, 17</td>
<td>51, 16</td>
</tr>
<tr>
<td>RELPRIME</td>
<td>Are $x$ and $y$ relatively prime?</td>
<td>Euclid (300 BCE)</td>
<td>34, 39</td>
<td>34, 51</td>
</tr>
<tr>
<td>PRIMES</td>
<td>Is $x$ prime?</td>
<td>AKS (2002)</td>
<td>53</td>
<td>51</td>
</tr>
<tr>
<td>EDIT-DISTANCE</td>
<td>Is the edit distance between $x$ and $y$ less than 5?</td>
<td>Dynamic programming</td>
<td>neither</td>
<td>acgggt</td>
</tr>
</tbody>
</table>

2. P, NP

- $P$: Decision problems for which there is a polynomial time algorithm.
- NP: Decision problems for which there is a polynomial time certifier.
**Independent Set**

**INDEPENDENT SET:** Given a graph \( G = (V, E) \) and an integer \( k \), is there a subset of vertices \( S \subseteq V \) such that \( |S| \geq k \), and for each edge at most one of its endpoints is in \( S \)?

What is the largest independent set?

Are the red nodes an independent set?

**There is no known polynomial algorithm to solve this.**
Are the red nodes an independent set?

There is a polynomial-time algorithm to check whether a set of nodes is an independent set for a graph.

Independent set is in NP

Claim: \( P \subseteq NP \).

Proof. Consider any problem \( X \) in \( P \).

By definition, there exists a polynomial time algorithm \( A() \) that solves \( X \).

Certificate: \( t = \epsilon \), certifier \( C(s, t) = A(s) \).
The Main Question: P Versus NP

Does $P = NP$?

Is the decision problem as easy as the certification problem?

Clay $1 million prize.

If yes: Efficient algorithms for 3-SAT, vertex cover, …

If no: No efficient algorithms possible for 3-SAT, vertex cover, …
More Hard Computational Problems

- Aerospace engineering: optimal mesh partitioning for finite elements.
- Biology: protein folding.
- Chemical engineering: heat exchanger network synthesis.
- Civil engineering: equilibrium of urban traffic flow.
- Economics: computation of arbitrage in financial markets with friction.
- Electrical engineering: VLSI layout.
- Environmental engineering: optimal placement of contaminant sensors.
- Financial engineering: find minimum risk portfolio of given return.
- Game theory: find Nash equilibrium that maximizes social welfare.
- Genomics: phylogeny reconstruction.
- Mechanical engineering: structure of turbulence in sheared flows.
- Medicine: reconstructing 3-D shape from biplane angiogram.
- Operations research: optimal resource allocation.
- Physics: partition function of 3-D Ising model in statistical mechanics.
- Politics: Shapley-Shubik voting power.
- Statistics: optimal experimental design.

Jon Kleinberg
Rebel King

Received 2008 ACM-Infosys Foundation Award in the Computing Sciences for his contributions to the science of networks and the World Wide Web which combine deep social insights and mathematical reasoning.

Received ACM's AAAI Allen Newell Award in 2014 for groundbreaking work in computer science in areas including social and information networks, information retrieval, and data science, and for bridging computing, economics and the social sciences.

Jon Kleinberg
ACM Fellow - 2013
For contributions to the science of information and social networks.

Eva Tardos
ACM Fellow - 1998
For fundamental contributions in the design and analysis of algorithms, combinatorial optimization, network flows, and approximation algorithms.
Applications of what we have learned

- **Graph theory:**
  - Used to study large networks found in physical, biological and social sciences
- **O()**
  - Do you suppose Google cares?
- **Greedy algorithms:**
  - Approximation for problems with no known efficient solutions
- **Divide and conquer:**
  - Computational geometry
  - Fast Fourier Transform - used to convert analog information to digital
- **Dynamic programming:**
  - Computational biology
  - Internet routing

Algorithms Timeline

- 1736 - Euler develops graph theory
- 1945 - mergesort
- 1956 - Kruskal's algorithm
- 1957 - Prim's algorithm
- 1957 - beginning of dynamic programming
- 1959 - Dijkstra shortest path
- 1961 - Segmented least squares
- 1962 - Ford-Fulkerson algorithm
- 1962 - Stable Matching Problem defined
- 1964 - Priority queues invented
  - early 1970s - O() concept formalized
  - early 1970s - NP-completeness defined
  - early 1970s - closest pair of points
  - early 1970s - string comparison

Algorithms Researchers who have won Turing Award

- 1972 - Edsger Dijkstra
- 1974 - Donald Knuth - analysis of algorithms
- 1976 - Michael Rabin - helped define P and NP
- 1978 - Robert Floyd - priority queues
- 1982 - Stephen Cook - NP-completeness
- 1985 - Richard Karp - network flow, NP-completeness
- 1986 - John Hopcroft, Robert Tarjan - analysis of algorithms
- 1993 - Richard Stearns, Juris Hartmanis - foundations of computational complexity theory
- 1995 - Manuel Blum - computational complexity theory
- 2000 - Andrew Yao - complexity theory
- 2012 - Silvio Micali and Shafi Goldwasser - cryptography, complexity theory
Topics we have not explored

- Problems not in NP: planning, games like chess
- Approximation algorithms
- Parallel algorithms

Will you ever use this material again?

- Other courses?
- Jobs?
- Grad school?
Please do course evaluations online!