

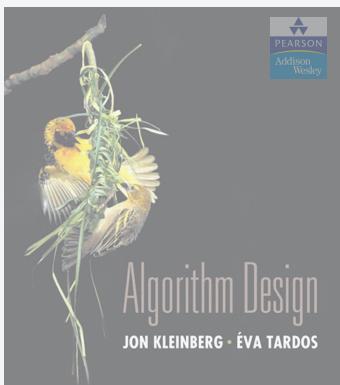
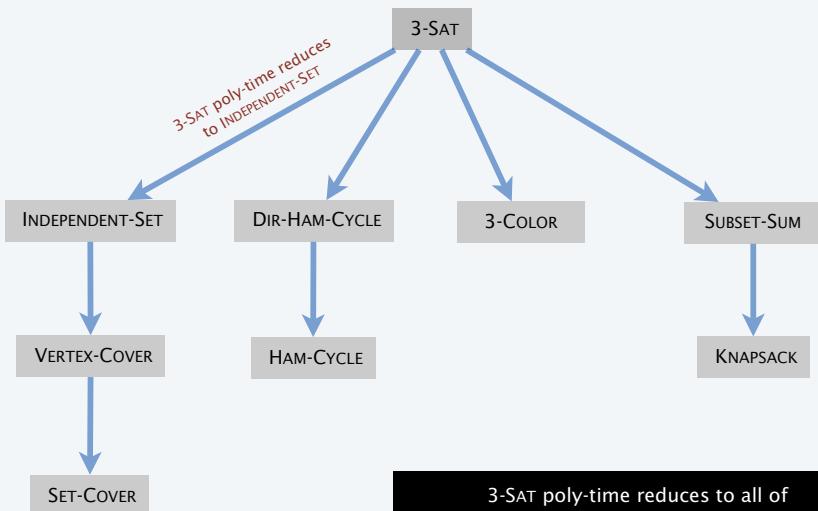
Lecture slides by Kevin Wayne  
Copyright © 2005 Pearson-Addison Wesley  
<http://www.cs.princeton.edu/~wayne/kleinberg-tardos>

Last updated on 5/5/18 5:20 AM

## 8. INTRACTABILITY II

- ▶  $P$  vs.  $NP$
- ▶  $NP$ -complete
- ▶ co- $NP$
- ▶  $NP$ -hard

## Recap



SECTION 8.3

## 8. INTRACTABILITY II

- ▶  $P$  vs.  $NP$
- ▶  $NP$ -complete
- ▶ co- $NP$
- ▶  $NP$ -hard

## P

### Decision problem.

- Problem  $X$  is a set of strings.
- Instance  $s$  is one string.
- Algorithm  $A$  solves problem  $X$ :  $A(s) = \begin{cases} \text{yes} & \text{if } s \in X \\ \text{no} & \text{if } s \notin X \end{cases}$

**Def.** Algorithm  $A$  runs in **polynomial time** if for every string  $s$ ,  $A(s)$  terminates in  $\leq p(|s|)$  “steps,” where  $p(\cdot)$  is some polynomial function.

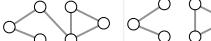
↑  
length of  $s$

**Def.** **P** = set of decision problems for which there exists a poly-time algorithm.

problem PRIMES:	{ 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, ... }
instance s:	592335744548702854681
algorithm:	Agrawal–Kayal–Saxena (2002)

## Some problems in P

P. Decision problems for which there exists a poly-time algorithm.

problem	description	poly-time algorithm	yes	no
MULTIPLE	Is $x$ a multiple of $y$ ?	grade-school division	51, 17	51, 16
REL-PRIME	Are $x$ and $y$ relatively prime?	Euclid's algorithm	34, 39	34, 51
PRIMES	Is $x$ prime?	Agrawal–Kayal–Saxena	53	51
EDIT-DISTANCE	Is the edit distance between $x$ and $y$ less than 5?	Needleman–Wunsch	neither neither	acgggt ttttta
L-SOLVE	Is there a vector $x$ that satisfies $Ax = b$ ?	Gauss–Edmonds elimination	$\begin{bmatrix} 0 & 1 & 1 \\ 2 & 4 & -2 \\ 0 & 3 & 15 \end{bmatrix}, \begin{bmatrix} 4 \\ 2 \\ 36 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$
U-CONN	Is an undirected graph $G$ connected?	depth-first search		

5

## NP

**Def.** Algorithm  $C(s, t)$  is a **certifier** for problem  $X$  if for every string  $s$ :  $s \in X$  iff there exists a string  $t$  such that  $C(s, t) = \text{yes}$ .

**Def.** **NP** = set of decision problems for which there exists a poly-time certifier.

- $C(s, t)$  is a poly-time algorithm.
- Certificate  $t$  is of polynomial size:  $|t| \leq p(|s|)$  for some polynomial  $p(\cdot)$ .

“certificate” or “witness”

problem COMPOSITES:	{ 4, 6, 8, 9, 10, 12, 14, 15, 16, 18, 20, .... }
instance s:	437669
certificate t:	541 ← 437,669 = 541 × 809
certifier C(s, t):	grade-school division

6

## Certifiers and certificates: satisfiability

**SAT.** Given a CNF formula  $\Phi$ , does it have a satisfying truth assignment?

**3-SAT.** SAT where each clause contains exactly 3 literals.

**Certificate.** An assignment of truth values to the Boolean variables.

**Certifier.** Check that each clause in  $\Phi$  has at least one true literal.

instance s	$\Phi = (\bar{x}_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee x_4)$
certificate t	$x_1 = \text{true}, x_2 = \text{true}, x_3 = \text{false}, x_4 = \text{false}$

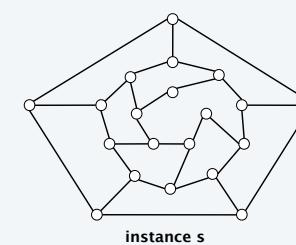
**Conclusions.** SAT ∈ NP, 3-SAT ∈ NP.

## Certifiers and certificates: Hamilton path

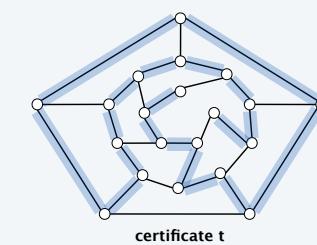
**HAMILTON-PATH.** Given an undirected graph  $G = (V, E)$ , does there exist a simple path  $P$  that visits every node?

**Certificate.** A permutation  $\pi$  of the  $n$  nodes.

**Certifier.** Check that  $\pi$  contains each node in  $V$  exactly once, and that  $G$  contains an edge between each pair of adjacent nodes.



**Conclusion.** HAMILTON-PATH ∈ NP.



7

8

## Some problems in NP

**NP.** Decision problems for which there exists a poly-time certifier.

problem	description	poly-time algorithm	yes	no
L-SOLVE	Is there a vector $x$ that satisfies $Ax = b$ ?	Gauss–Edmonds elimination	$\begin{bmatrix} 0 & 1 & 1 \\ 2 & 4 & -2 \\ 0 & 3 & 15 \end{bmatrix}, \begin{bmatrix} 4 \\ 2 \\ 36 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$
COMPOSITES	Is $x$ composite?	Agrawal–Kayal–Saxena	51	53
FACTOR	Does $x$ have a nontrivial factor less than $y$ ?	???	(56159, 50)	(55687, 50)
SAT	Given a CNF formula, does it have a satisfying truth assignment?	???	$\neg x_1 \vee x_2 \vee \neg x_3$ $x_1 \vee \neg x_2 \vee x_3$ $\neg x_1 \vee \neg x_2 \vee x_3$	$\neg x_2$ $x_1 \vee x_2$ $\neg x_1 \vee x_2$
HAMILTON-PATH	Is there a simple path between $u$ and $v$ that visits every node?	???		

9



## Intractability: quiz 1

Which of the following graph problems are known to be in NP?

- A. Is the length of the longest simple path  $\leq k$ ?
- B. Is the length of the longest simple path  $\geq k$ ?
- C. Is the length of the longest simple path  $= k$ ?
- D. Find the length of the longest simple path.
- E. All of the above.

10

## Intractability: quiz 2



In complexity theory, the abbreviation NP stands for...

- A. Nope.
- B. No problem.
- C. Not polynomial time.
- D. Not polynomial space.
- E. Nondeterministic polynomial time.

11

## Significance of NP

**NP.** Decision problems for which there exists a poly-time certifier.

“NP captures vast domains of computational, scientific, and mathematical endeavors, and seems to roughly delimit what mathematicians and scientists have been aspiring to compute feasibly.” — Christos Papadimitriou

“In an ideal world it would be renamed P vs VP.” — Clyde Kruskal

12

## P, NP, and EXP

- P. Decision problems for which there exists a poly-time algorithm.
- NP. Decision problems for which there exists a poly-time certifier.
- EXP. Decision problems for which there exists an exponential-time algorithm.

**Proposition.**  $P \subseteq NP$ .

**Pf.** Consider any problem  $X \in P$ .

- By definition, there exists a poly-time algorithm  $A(s)$  that solves  $X$ .
- Certificate  $t = \varepsilon$ , certifier  $C(s, t) = A(s)$ . ■

**Proposition.**  $NP \subseteq EXP$ .

**Pf.** Consider any problem  $X \in NP$ .

- By definition, there exists a poly-time certifier  $C(s, t)$  for  $X$ , where certificate  $t$  satisfies  $|t| \leq p(|s|)$  for some polynomial  $p(\cdot)$ .
- To solve instance  $s$ , run  $C(s, t)$  on all strings  $t$  with  $|t| \leq p(|s|)$ .
- Return yes iff  $C(s, t)$  returns yes for any of these potential certificates. ■

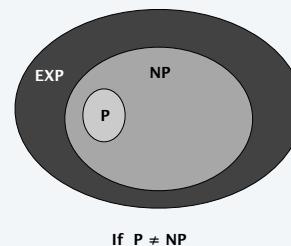
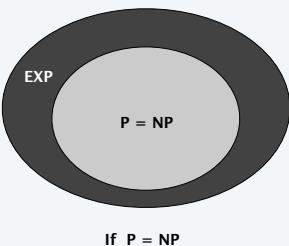
**Fact.**  $P \neq EXP \Rightarrow$  either  $P \neq NP$ , or  $NP \neq EXP$ , or both.

13

## The main question: P vs. NP

Does  $P = NP$ ? [Cook 1971, Edmonds, Levin, Yablonski, Gödel]

Is the decision problem as easy as the certification problem?



If yes... Efficient algorithms for 3-SAT, TSP, VERTEX-COVER, FACTOR, ...

If no... No efficient algorithms possible for 3-SAT, TSP, VERTEX-COVER, ...

**Consensus opinion.** Probably no.

## The main question: P vs. NP

Q. How to solve an instance of 3-SAT with  $n$  variables?

A. Exhaustive search: try all  $2^n$  truth assignments.

Q. Can we do anything substantially more clever?

**Conjecture.** No poly-time algorithm for 3-SAT.

"intractable"



14

## Reductions: quiz 3



Suppose  $P \neq NP$ . Which of the following are still possible?

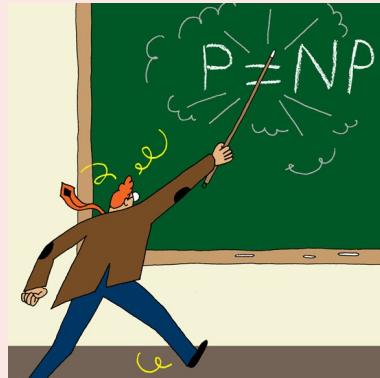
- A.  $O(n^3)$  algorithm for factoring  $n$ -bit integers.
- B.  $O(1.657^n)$  time algorithm for HAMILTON-CYCLE.
- C.  $O(n^{\log \log \log n})$  algorithm for 3-SAT.
- D. All of the above.

15

16

Does  $P = NP$ ?

- A. Yes.
- B. No.
- C. None of the above.



17

## Possible outcomes

 $P \neq NP$ 

*"We seem to be missing even the most basic understanding of the nature of its difficulty.... All approaches tried so far probably (in some cases, provably) have failed. In this sense  $P = NP$  is different from many other major mathematical problems on which a gradual progress was being constantly done (sometimes for centuries) whereupon they yielded, either completely or partially."*

— Alexander Razborov (2002)



19

## Possible outcomes

 $P \neq NP$ 

*"I conjecture that there is no good algorithm for the traveling salesman problem. My reasons are the same as for any mathematical conjecture: (i) It is a legitimate mathematical possibility and (ii) I do not know."*

— Jack Edmonds 1966



*"In my view, there is no way to even make intelligent guesses about the answer to any of these questions. If I had to bet now, I would bet that  $P$  is not equal to  $NP$ . I estimate the half-life of this problem at 25–50 more years, but I wouldn't bet on it being solved before 2100."*

— Bob Tarjan (2002)



18

## Possible outcomes

 $P = NP$ 

*"I think that in this respect I am on the loony fringe of the mathematical community: I think (not too strongly!) that  $P = NP$  and this will be proved within twenty years. Some years ago, Charles Read and I worked on it quite bit, and we even had a celebratory dinner in a good restaurant before we found an absolutely fatal mistake."*

— Béla Bollobás (2002)



*"In my opinion this shouldn't really be a hard problem; it's just that we came late to this theory, and haven't yet developed any techniques for proving computations to be hard. Eventually, it will just be a footnote in the books."* — John Conway



20

## Other possible outcomes

$P = NP$ , but only  $\Omega(n^{100})$  algorithm for 3-SAT.

$P \neq NP$ , but with  $O(n^{\log^* n})$  algorithm for 3-SAT.

$P = NP$  is independent (of ZFC axiomatic set theory).

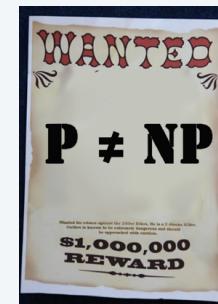
*“It will be solved by either 2048 or 4096. I am currently somewhat pessimistic. The outcome will be the truly worst case scenario: namely that someone will prove  $P = NP$  because there are only finitely many obstructions to the opposite hypothesis; hence there exists a polynomial time solution to SAT but we will never know its complexity!” — Donald Knuth*



21

## Millennium prize

Millennium prize. \$1 million for resolution of  $P \neq NP$  problem.



**Clay Mathematics Institute**  
Dedicated to increasing and disseminating mathematical knowledge

[HOME](#) | [ABOUT CMI](#) | [PROGRAMS](#) | [NEWS & EVENTS](#) | [AWARDS](#) | [SCHOLARS](#) | [PUBLICATIONS](#)

**Millennium Problems**

In order to celebrate mathematics in the new millennium, The Clay Mathematics Institute of Cambridge, Massachusetts (CMI) has named seven Prize Problems. The Scientific Advisory Board of CMI selected these problems, focusing on important classic questions that have resisted solution over the years. The Board of Directors of CMI designated a \$7 million prize fund for the solution to these problems, with \$1 million allocated to each. During the Millennium Meeting held on May 24, at the Institut des Hautes Études Scientifiques in Paris, France, the Clay Mathematics Institute invited several mathematicians to give lectures on the problems. The general public, while John Tate and Michael Atiyah spoke on the problems. The CMI invited specialists to formulate each problem.

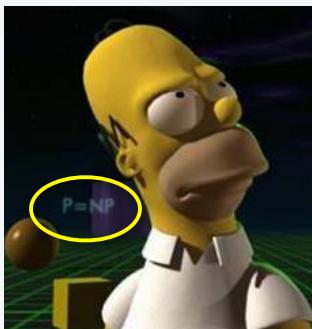
[Birch and Swinnerton-Dyer Conjecture](#)  
[Hodge Conjecture](#)  
[Navier-Stokes Equations](#)  
[P vs NP](#)  
[Poincaré Conjecture](#)  
[Riemann Hypothesis](#)  
[Yang-Mills Theory](#)  
[Rules](#)  
[Millennium Meeting Videos](#)

22

## NP-completeness and pop culture

### Some writers for the Simpsons and Futurama.

- J. Steward Burns. *M.S. in mathematics (Berkeley '93)*.
- David X. Cohen. *M.S. in computer science (Berkeley '92)*.
- Al Jean. *B.S. in mathematics. (Harvard '81)*.
- Ken Keeler. *Ph.D. in applied mathematics (Harvard '90)*.
- Jeff Westbrook. *Ph.D. in computer science (Princeton '89)*.



Copyright © 1990, Matt Groening



Copyright © 2000, Twentieth Century Fox

23



char	ASCII	binary
P	80	1010000
=	61	0111101
N	78	1001110
P	80	1010000
?	63	0111111

25

## Polynomial transformations

**Def.** Problem  $X$  **polynomial (Cook)** reduces to problem  $Y$  if arbitrary instances of problem  $X$  can be solved using:

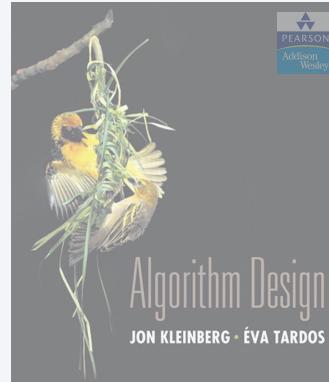
- Polynomial number of standard computational steps, plus
- Polynomial number of calls to oracle that solves problem  $Y$ .

**Def.** Problem  $X$  **polynomial (Karp)** transforms to problem  $Y$  if given any instance  $x$  of  $X$ , we can construct an instance  $y$  of  $Y$  such that  $x$  is a *yes* instance of  $X$  iff  $y$  is a *yes* instance of  $Y$ .  
 ↑  
 we require  $|y|$  to be of size polynomial in  $|x|$

**Note.** Polynomial transformation is polynomial reduction with just one call to oracle for  $Y$ , exactly at the end of the algorithm for  $X$ . Almost all previous reductions were of this form.

**Open question.** Are these two concepts the same with respect to **NP**?  
 ↑  
 we abuse notation  $\leq_p$  and blur distinction

27



SECTION 8.4

## 8. INTRACTABILITY II

- ▶  $P$  vs.  $NP$
- ▶  $NP$ -complete
- ▶  $co\text{-}NP$
- ▶  $NP$ -hard

## NP-complete

**NP-complete.** A problem  $Y \in NP$  with the property that for every problem  $X \in NP$ ,  $X \leq_p Y$ .

**Proposition.** Suppose  $Y \in NP$ -complete. Then,  $Y \in P$  iff  $P = NP$ .

**Pf.**  $\Leftarrow$  If  $P = NP$ , then  $Y \in P$  because  $Y \in NP$ .

**Pf.**  $\Rightarrow$  Suppose  $Y \in P$ .

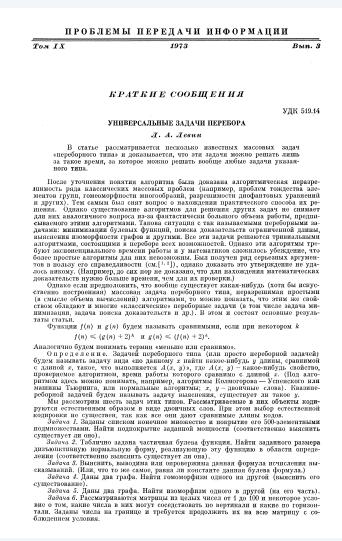
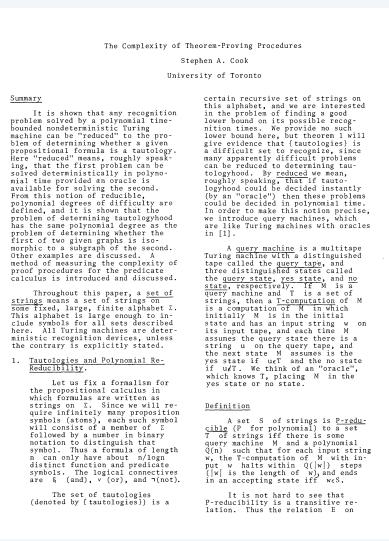
- Consider any problem  $X \in NP$ . Since  $X \leq_p Y$ , we have  $X \in P$ .
- This implies  $NP \subseteq P$ .
- We already know  $P \subseteq NP$ . Thus  $P = NP$ . ▀

**Fundamental question.** Are there any “natural”  $NP$ -complete problems?

28

## The “first” NP-complete problem

**Theorem.** [Cook 1971, Levin 1973]  $SAT \in \text{NP}\text{-complete}$ .



29

## Establishing NP-completeness

**Remark.** Once we establish first “natural” NP-complete problem, others fall like dominoes.

**Recipe.** To prove that  $Y \in \text{NP}\text{-complete}$ :

- Step 1. Show that  $Y \in \text{NP}$ .
- Step 2. Choose an NP-complete problem  $X$ .
- Step 3. Prove that  $X \leq_p Y$ .

**Proposition.** If  $X \in \text{NP}\text{-complete}$ ,  $Y \in \text{NP}$ , and  $X \leq_p Y$ , then  $Y \in \text{NP}\text{-complete}$ .

**Pf.** Consider any problem  $W \in \text{NP}$ . Then, both  $W \leq_p X$  and  $X \leq_p Y$ .

- By transitivity,  $W \leq_p Y$ .
- Hence  $Y \in \text{NP}\text{-complete}$ .

↑  
by definition of  
NP-complete      ↑  
by assumption

## Reductions: quiz 4

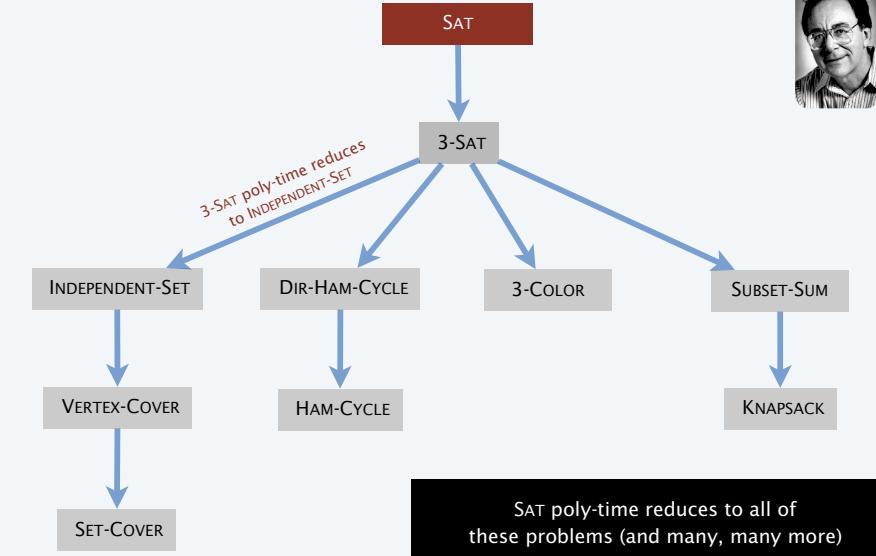


Suppose that  $X \in \text{NP-COMPLETE}$ ,  $Y \in \text{NP}$ , and  $X \leq_p Y$ . Which can you infer?

- A.  $Y$  is NP-complete.
- B. If  $Y \notin \text{P}$ , then  $\text{P} \neq \text{NP}$ .
- C. If  $\text{P} \neq \text{NP}$ , then neither  $X$  nor  $Y$  is in  $\text{P}$ .
- D. All of the above.

31

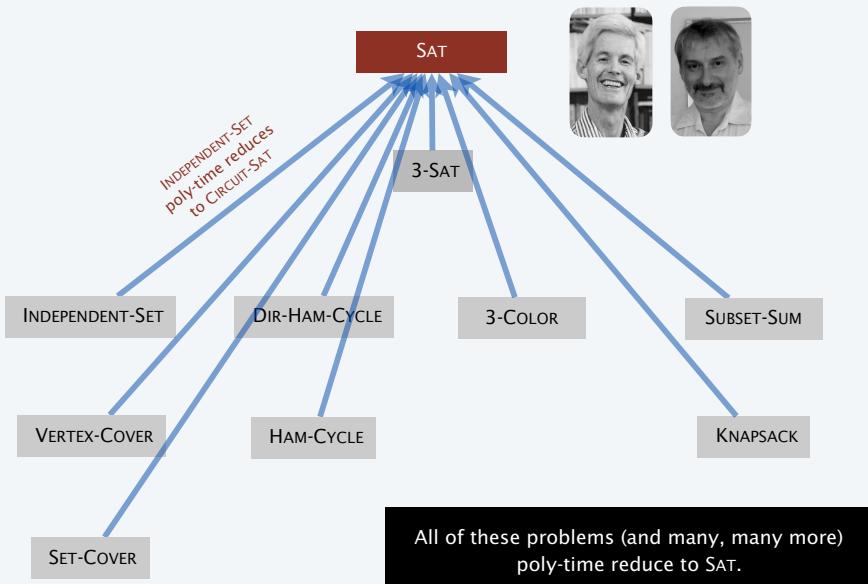
## Implications of Karp



30

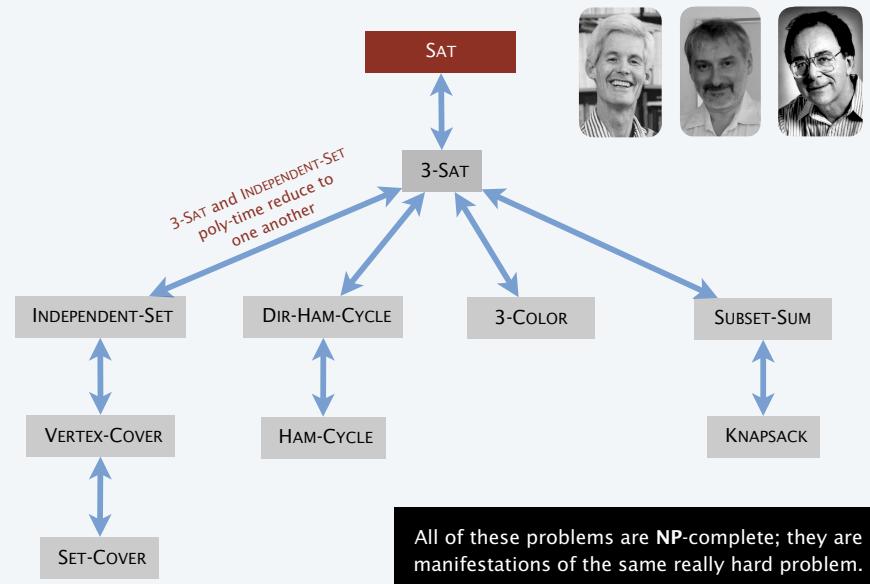
32

## Implications of Cook–Levin



33

## Implications of Karp + Cook–Levin



34

I'D TELL YOU ANOTHER  
NP-COMPLETE JOKE,  
BUT ONCE YOU'VE HEARD ONE,  
  
YOU'VE HEARD THEM ALL.

## Some NP-complete problems

### Basic genres of NP-complete problems and paradigmatic examples.

- Packing/covering problems: SET-COVER, VERTEX-COVER, INDEPENDENT-SET.
- Constraint satisfaction problems: CIRCUIT-SAT, SAT, 3-SAT.
- Sequencing problems: HAMILTON-CYCLE, TSP.
- Partitioning problems: 3D-MATCHING, 3-COLOR.
- Numerical problems: SUBSET-SUM, KNAPSACK.

**Practice.** Most NP problems are known to be in either P or NP-complete.

**NP-intermediate?** FACTOR, DISCRETE-LOG, GRAPH-ISOMORPHISM, ....

**Theorem.** [Ladner 1975] Unless P = NP, there exist problems in NP that are in neither P nor NP-complete.

On the Structure of Polynomial Time Reducibility  
RICHARD E. LADNER  
University of Washington, Seattle, Washington

36

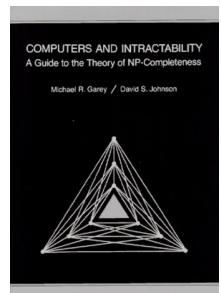
## More hard computational problems

Garey and Johnson. Computers and Intractability.

- Appendix includes over 300 NP-complete problems.
- Most cited reference in computer science literature.

### Most Cited Computer Science Citations

This list is generated from documents in the CiteSeer <sup>x</sup> database as of January 17, 2013. This list is automatically generated and may contain errors. The list is generated in batch mode and citation counts may differ from those currently in the CiteSeer <sup>x</sup> database, since the database is continuously updated.	
All Years   1990   1991   1992   1993   1994   1995   1996   1997   1998   1999   2000   2001   2002   2003   2004   2005   2006   2007   2008   2009   2010   2011   2012   2013	
1. M R Garey, D S Johnson Computers and Intractability. A Guide to the Theory of NP-Completeness 1979 8665	
2. T Cormen, C E Leiserson, R Rivest Introduction to Algorithms 1990 7210	
3. V N Vapnik The nature of statistical learning theory 1998 6580	
4. A P Dempster, N M Laird, D B Rubin Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, 1977 6082	
5. J Cover, J Thomas Elements of Information Theory 1991 6075	
6. D E Goldberg Genetic Algorithms in Search, Optimization, and Machine Learning, 1989 5998	
7. J Pearl Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference 1988 5592	
8. E Gamma, R Helm, R Johnson, J Vissides Design Patterns: Elements of Reusable Object-Oriented Software 1995 4614	
9. C E Shannon A mathematical theory of communication Bell Syst. Tech. J. 1948 4118	
10. J R Quinlan C4.5: Programs for Machine Learning 1993 4018	



37

## More hard computational problems

Aerospace engineering. Optimal mesh partitioning for finite elements.

Biology. Phylogeny reconstruction.

Chemical engineering. Heat exchanger network synthesis.

Chemistry. Protein folding.

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. Minimum risk portfolio of given return.

Game theory. Nash equilibrium that maximizes social welfare.

Mathematics. Given integer  $a_1, \dots, a_n$ , compute  $\int_0^{2\pi} \cos(a_1\theta) \times \cos(a_2\theta) \times \dots \times \cos(a_n\theta) d\theta$

Mechanical engineering. Structure of turbulence in sheared flows.

Medicine. Reconstructing 3d shape from biplane angiogram.

Operations research. Traveling salesperson problem.

Physics. Partition function of 3d Ising model.

Politics. Shapley–Shubik voting power.

Recreation. Versions of Sudoku, Checkers, Minesweeper, Tetris, Rubik's Cube.

Statistics. Optimal experimental design.

38

## Extent and impact of NP-completeness

Extent of NP-completeness. [Papadimitriou 1995]

- Prime intellectual export of CS to other disciplines.
- 6,000 citations per year (more than “compiler”, “OS”, “database”).
- Broad applicability and classification power.

NP-completeness can guide scientific inquiry.

- 1926: Ising introduces simple model for phase transitions.
- 1944: Onsager finds closed-form solution to 2D-ISING in tour de force.
- 19xx: Feynman and other top minds seek solution to 3D-ISING.
- 2000: Istrail proves 3D-ISING  $\in$  NP-complete.
  - search for closed formula appears doomed
  - a holy grail of statistical mechanics

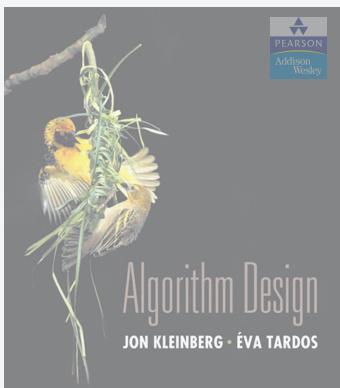


39

## You NP-complete me



40



SECTION 8.9

## 8. INTRACTABILITY II

- ▶ *P vs. NP*
- ▶ *NP-complete*
- ▶ *co-NP*
- ▶ *NP-hard*

### Asymmetry of NP

**Asymmetry of NP.** We need short certificates only for *yes* instances.

**Ex 1.** SAT vs. UN-SAT.

- Can prove a CNF formula is satisfiable by specifying an assignment.
- How could we prove that a formula is not satisfiable?

**SAT.** Given a CNF formula  $\Phi$ , is there a satisfying truth assignment?

**UN-SAT.** Given a CNF formula  $\Phi$ , is there no satisfying truth assignment?

### Asymmetry of NP

**Asymmetry of NP.** We need short certificates only for *yes* instances.

**Ex 2.** HAMILTON-CYCLE vs. No-HAMILTON-CYCLE.

- Can prove a graph is Hamiltonian by specifying a permutation.
- How could we prove that a graph is not Hamiltonian?

**HAMILTON-CYCLE.** Given a graph  $G = (V, E)$ , is there a simple cycle  $\Gamma$  that contains every node in  $V$ ?

**No-HAMILTON-CYCLE.** Given a graph  $G = (V, E)$ , is there no simple cycle  $\Gamma$  that contains every node in  $V$ ?

### Asymmetry of NP

**Asymmetry of NP.** We need short certificates only for *yes* instances.

**Q.** How to classify UN-SAT and No-HAMILTON-CYCLE ?

- SAT  $\in \text{NP}$ -complete and SAT  $\equiv_P$  UN-SAT.
- HAMILTON-CYCLE  $\in \text{NP}$ -complete and HAMILTON-CYCLE  $\equiv_P$  No-HAMILTON-CYCLE.
- But neither UN-SAT nor No-HAMILTON-CYCLE are known to be in **NP**.

## NP and co-NP

**NP.** Decision problems for which there is a poly-time certifier.

**Ex.** SAT, HAMILTON-CYCLE, and COMPOSITES.

**Def.** Given a decision problem  $X$ , its **complement**  $\overline{X}$  is the same problem with the *yes* and *no* answers reversed.

**Ex.**  $X = \{ 4, 6, 8, 9, 10, 12, 14, 15, \dots \}$   $\overleftarrow{\text{ignore 0 and 1}}$   
(neither prime nor composite)

**co-NP.** Complements of decision problems in **NP**.

**Ex.** UN-SAT, NO-HAMILTON-CYCLE, and PRIMES.

## NP = co-NP ?

**Fundamental open question.** Does **NP = co-NP**?

- Do *yes* instances have succinct certificates iff *no* instances do?
- Consensus opinion: no.

**Theorem.** If  $\mathbf{NP} \neq \mathbf{co-NP}$ , then  $\mathbf{P} \neq \mathbf{NP}$ .

**Pf idea.**

- **P** is closed under complementation.
- If  $\mathbf{P} = \mathbf{NP}$ , then **NP** is closed under complementation.
- In other words,  $\mathbf{NP} = \mathbf{co-NP}$ .
- This is the contrapositive of the theorem.

45

## Good characterizations

**Good characterization.** [Edmonds 1965]  $\mathbf{NP} \cap \mathbf{co-NP}$ .

- If problem  $X$  is in both **NP** and **co-NP**, then:
  - for *yes* instance, there is a succinct certificate
  - for *no* instance, there is a succinct disqualifier
- Provides conceptual leverage for reasoning about a problem.

**Ex.** Given a bipartite graph, is there a perfect matching?

- If yes, can exhibit a perfect matching.
- If no, can exhibit a set of nodes  $S$  such that  $|neighbors(S)| < |S|$ .

JOURNAL OF RESEARCH of the National Bureau of Standards—B. Mathematics and Mathematical Physics  
Vol. 69B, Nos. 1 and 2, January-June 1965

Minimum Partition of a Matroid Into Independent Subsets

Jack Edmonds

(December 1, 1964)

A matroid  $M$  is a finite set  $M$  of elements with a family of subsets, called independent, such that (1) every subset of an independent set is independent, and (2) for every subset  $A$  of  $M$ , all maximal independent subsets of  $A$  have the same cardinality, called the rank  $r(A)$  of  $A$ . It is proved that a matroid can be partitioned into as few as  $k$  sets, each independent, if and only if every subset  $A$  has cardinality at most  $k - r(A)$ .

46

## Good characterizations

We seek a good characterization of the minimum number of independent sets into which the columns of a matrix of  $M_F$  can be partitioned. As the criterion of “good” for the characterization we apply the “principle of the absolute supervisor.” The good characterization will describe certain information about the matrix which the supervisor can require his assistant to search out along with a minimum partition and which the supervisor can then use with ease to verify with mathematical certainty that the partition is indeed minimum. Having a good characterization does not mean necessarily that there is a good algorithm. The assistant might have to kill himself with work to find the information and the partition.

47

48

## Good characterizations

**Observation.**  $\mathbf{P} \subseteq \mathbf{NP} \cap \text{co-NP}$ .

- Proof of max-flow min-cut theorem led to stronger result that max-flow and min-cut are in  $\mathbf{P}$ .
- Sometimes finding a good characterization seems easier than finding an efficient algorithm.

**Fundamental open question.** Does  $\mathbf{P} = \mathbf{NP} \cap \text{co-NP}$ ?

- Mixed opinions.
- Many examples where problem found to have a nontrivial good characterization, but only years later discovered to be in  $\mathbf{P}$ .

49

## Linear programming is in $\mathbf{NP} \cap \text{co-NP}$

**LINEAR-PROGRAMMING.** Given  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $c \in \mathbb{R}^n$ , and  $\alpha \in \mathbb{R}$ , does there exist  $x \in \mathbb{R}^n$  such that  $Ax \leq b$ ,  $x \geq 0$  and  $c^T x \geq \alpha$ ?

**Theorem.** [Gale–Kuhn–Tucker 1948] LINEAR-PROGRAMMING  $\in \mathbf{NP} \cap \text{co-NP}$ .

**Pf sketch.** If (P) and (D) are nonempty, then  $\max = \min$ .

$$\begin{array}{ll} (\text{P}) & \max c^T x \\ \text{s. t.} & Ax \leq b \\ & x \geq 0 \end{array} \quad \begin{array}{ll} (\text{D}) & \min y^T b \\ \text{s. t.} & A^T y \geq c \\ & y \geq 0 \end{array}$$

### CHAPTER XIX

#### LINEAR PROGRAMMING AND THE THEORY OF GAMES<sup>1</sup>

BY DAVID GALE, HAROLD W. KUHN, AND ALBERT W. TUCKER<sup>2</sup>

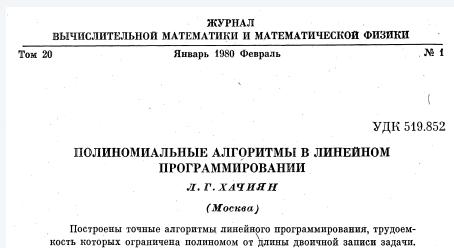
The basic “scalar” problem of linear programming is to maximize (or minimize) a linear function of several variables constrained by a system of linear inequalities [Dantzig, II]. A more general “vector” problem calls for maximizing (in a sense of partial order) a system of linear functions of several variables subject to a system of linear inequalities and, perhaps, linear equations [Koopmans, III]. The purpose of this chapter is to establish theorems of duality and existence for general “matrix” problems of linear programming which contain the “scalar” and “vector” problems as special cases, and to relate these general problems to the theory of zero-sum two-person games.

50

## Linear programming is in $\mathbf{NP} \cap \text{co-NP}$

**LINEAR-PROGRAMMING.** Given  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $c \in \mathbb{R}^n$ , and  $\alpha \in \mathbb{R}$ , does there exist  $x \in \mathbb{R}^n$  such that  $Ax \leq b$ ,  $x \geq 0$  and  $c^T x \geq \alpha$ ?

**Theorem.** [Khachiyan 1979] LINEAR-PROGRAMMING  $\in \mathbf{P}$ .



51

## Primality testing is in $\mathbf{NP} \cap \text{co-NP}$

**Theorem.** [Pratt 1975] PRIMES  $\in \mathbf{NP} \cap \text{co-NP}$ .

SIAM J. COMPUT.  
Vol. 4, No. 3, September 1975

### EVERY PRIME HAS A SUCCINCT CERTIFICATE\*

VAUGHAN R. PRATT†

**Abstract.** To prove that a number  $n$  is composite, it suffices to exhibit the working for the multiplication of a pair of factors. This working, represented as a string, is of length bounded by a polynomial in  $\log n$ . We show that the same property holds for the primes. It is noteworthy that almost no other set is known to have the property that short proofs for membership or nonmembership exist for all candidates without being known to have the property that such proofs are easy to come by. It remains an open problem whether a prime  $n$  can be recognized in only  $\log_2 n$  operations of a Turing machine for any fixed  $\epsilon$ .

The proof system used for certifying primes is as follows.

AXIOM.  $(x, y, 1)$ .

INFERENCE RULES.

R<sub>1</sub>:  $(p, x, a), q \vdash (p, x, qa)$  provided  $x^{(p-1)/q} \not\equiv 1 \pmod p$  and  $q|(p-1)$ .

R<sub>2</sub>:  $(p, x, p-1) \vdash p$  provided  $x^{p-1} \equiv 1 \pmod p$ .

THEOREM 1.  $p$  is a theorem  $\equiv p$  is a prime.

THEOREM 2.  $p$  is a theorem  $\Rightarrow p$  has a proof of  $\lceil 4 \log_2 p \rceil$  lines.

52

## Primality testing is in NP ∩ co-NP

**Theorem.** [Pratt 1975] PRIMES ∈ NP ∩ co-NP.

Pf sketch. An odd integer  $s$  is prime iff there exists an integer  $1 < t < s$  s.t.

$$\begin{aligned} t^{s-1} &\equiv 1 \pmod{s} \\ t^{(s-1)/p} &\not\equiv 1 \pmod{s} \end{aligned}$$

for all prime divisors  $p$  of  $s-1$

instance **s** 437677  
certificate **t** 17,  $2^2 \times 3 \times 36473$

prime factorization of  $s-1$   
also need a recursive certificate  
to assert that 3 and 36,473 are prime

### CERTIFIER ( $s$ )

**CHECK**  $s-1 = 2 \times 2 \times 3 \times 36473$ .

**CHECK**  $17^{s-1} \equiv 1 \pmod{s}$ .

**CHECK**  $17^{(s-1)/2} \equiv 437676 \pmod{s}$ .

**CHECK**  $17^{(s-1)/3} \equiv 329415 \pmod{s}$ .

**CHECK**  $17^{(s-1)/36473} \equiv 305452 \pmod{s}$ .

use repeated squaring

53

## Factoring is in NP ∩ co-NP

**FACTORIZER.** Given an integer  $x$ , find its prime factorization.

**FACTOR.** Given two integers  $x$  and  $y$ , does  $x$  have a nontrivial factor  $< y$ ?

**Theorem.** FACTOR ≈<sub>P</sub> FACTORIZER.

Pf.

- $\leq_P$  trivial.
- $\geq_P$  binary search to find a factor; divide out the factor and repeat. ▀

**Theorem.** FACTOR ∈ NP ∩ co-NP.

Pf.

- Certificate: a factor  $p$  of  $x$  that is less than  $y$ .
- Disqualifier: the prime factorization of  $x$  (where each prime factor is less than  $y$ ), along with a Pratt certificate that each factor is prime. ▀

55

## Primality testing is in P

**Theorem.** [Agrawal–Kayal–Saxena 2004] PRIMES ∈ P.

Annals of Mathematics, 160 (2004), 781–793

### PRIMES is in P

By MANINDRA AGRAWAL, NEERAJ KAYAL, and NITIN SAXENA\*

#### Abstract

We present an unconditional deterministic polynomial-time algorithm that determines whether an input number is prime or composite.

54

## Is factoring in P?

**Fundamental question.** Is FACTOR ∈ P?

**Challenge.** Factor this number.

74037563479561712828046796097429573142593188889231289  
08493623263897276503402826627689199641962511784399589  
43305021275853701189680982867331732731089309005525051  
16877063299072396380786710086096962537934650563796359

RSA-704  
(\$30,000 prize if you can factor)

56

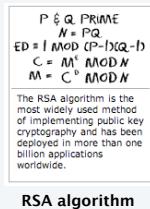
## Exploiting intractability

### Modern cryptography.

- Ex. Send your credit card to Amazon.
- Ex. Digitally sign an e-document.
- Enables freedom of privacy, speech, press, political association.

**RSA.** Based on dichotomy between complexity of two problems.

- To use: generate two random  $n$ -bit primes and multiply.
- To break: suffices to factor a  $2n$ -bit integer.



RSA algorithm



RSA sold  
for \$2.1 billion

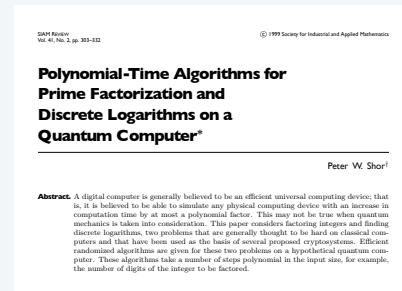


or design a t-shirt

57

## Factoring on a quantum computer

**Theorem.** [Shor 1994] Can factor an  $n$ -bit integer in  $O(n^3)$  steps on a “quantum computer.”



58

2001. Factored  $15 = 3 \times 5$  (with high probability) on a quantum computer.

2012. Factored  $21 = 3 \times 7$ .

**Fundamental question.** Does  $P = BQP$ ?

quantum analog of P  
(bounded error quantum polynomial time)

## 8. INTRACTABILITY II



- ▶  $P$  vs.  $NP$
- ▶  $NP$ -complete
- ▶  $co$ - $NP$
- ▶  $NP$ -hard

### A note on terminology

SIGACT News

12

January 1974

#### A TERMINOLOGICAL PROPOSAL

D. F. Knuth

While preparing a book on combinatorial algorithms, I felt a strong need for a new technical term, a word which is essentially a one-sided version of polynomial complete. A great many problems of practical interest have the property that they are at least as difficult to solve in polynomial time as those of the Cook-Karp class NP. I needed an adjective to convey such a degree of difficulty, both formally and informally; and since the range of practical applications is so broad, I felt it would be best to establish such a term as soon as possible.

The goal is to find an adjective  $x$  that sounds good in sentences like this:

The covering problem is  $x$ .  
It is  $x$  to decide whether a given graph has a Hamiltonian circuit.  
It is unknown whether or not primality testing is an  $x$  problem.

**Note.** The term  $x$  does not necessarily imply that a problem is in  $NP$ , just that every problem in  $NP$  poly-time reduces to  $x$ .

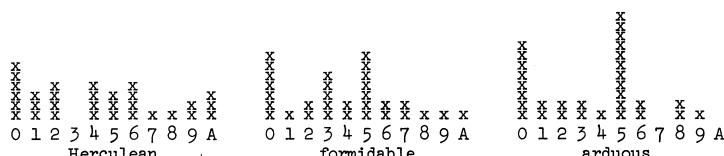
60

## A note on terminology

### Knuth's original suggestions.

- Hard.
- Tough.
- Herculean.
- Formidable.
- Arduous.

so common that it is unclear whether  
it is being used in a technical sense



assign a real number between 0 and 1 to each choice

61

## A note on terminology

### Some English word write-ins.

- Impractical.
- Bad.
- Heavy.
- Tricky.
- Intricate.
- Prodigious.
- Difficult.
- Intractable.
- Costly.
- Obdurate.
- Obstinate.
- Exorbitant.
- Interminable.

## A note on terminology

**Hard-boiled.** [Ken Steiglitz] In honor of Cook.

**Hard-ass.** [Al Meyer] Hard as satisfiability.

**Sisyphean.** [Bob Floyd] Problem of Sisyphus was time-consuming.

**Ulyssean.** [Donald Knuth] Ulysses was known for his persistence.

*"creative research workers are as full of ideas for new terminology  
as they are empty of enthusiasm for adopting it."*

— Donald Knuth

63

62

## A note on terminology: acronyms

**PET.** [Shen Lin] Probably exponential time.

- If  $P \neq NP$ , provably exponential time.
- If  $P = NP$ , previously exponential time.

**GNP.** [Al Meyer] Greater than or equal to  $NP$  in difficulty.

- And costing more than the GNP to solve.

64

## A note on terminology: made-up words

---

**Exparent.** [Mike Paterson] Exponential + apparent.

**Perarduous.** [Mike Paterson] Throughout (in space or time) + completely.

**Supersat.** [AI Meyer] Greater than or equal to satisfiability.

**Polychronious.** [Ed Reingold] Enduringly long; chronic.

## A note on terminology: consensus

---

**NP-complete.** A problem in **NP** such that every problem in **NP** poly-time reduces to it.

**NP-hard.** [Bell Labs, Steve Cook, Ron Rivest, Sartaj Sahni]

A problem such that every problem in **NP** poly-time reduces to it.

One final criticism (which applies to all the terms suggested) was stated nicely by Vaughan Pratt: "If the Martians know that  $P = NP$  for Turing Machines and they kidnap me, I would lose face calling these problems 'formidable'." Yes; if  $P = NP$ , there's no need for any term at all. But I'm willing to risk such an embarrassment, and in fact I'm willing to give a prize of one live turkey to the first person who proves that  $P = NP$ .