Al Aho

aho@cs.columbia.edu

The Quintessential Questions of Computer Science

Bell Labs
Murray Hill, NJ
June 22, 2011

Quintessence

From Webster's *New World Dictionary*

- **the pure, concentrated essence of anything**

Seventy-five Years of Computer Science

- What is the biggest impact that Computer Science has had on the world?

- Many would answer: the Internet

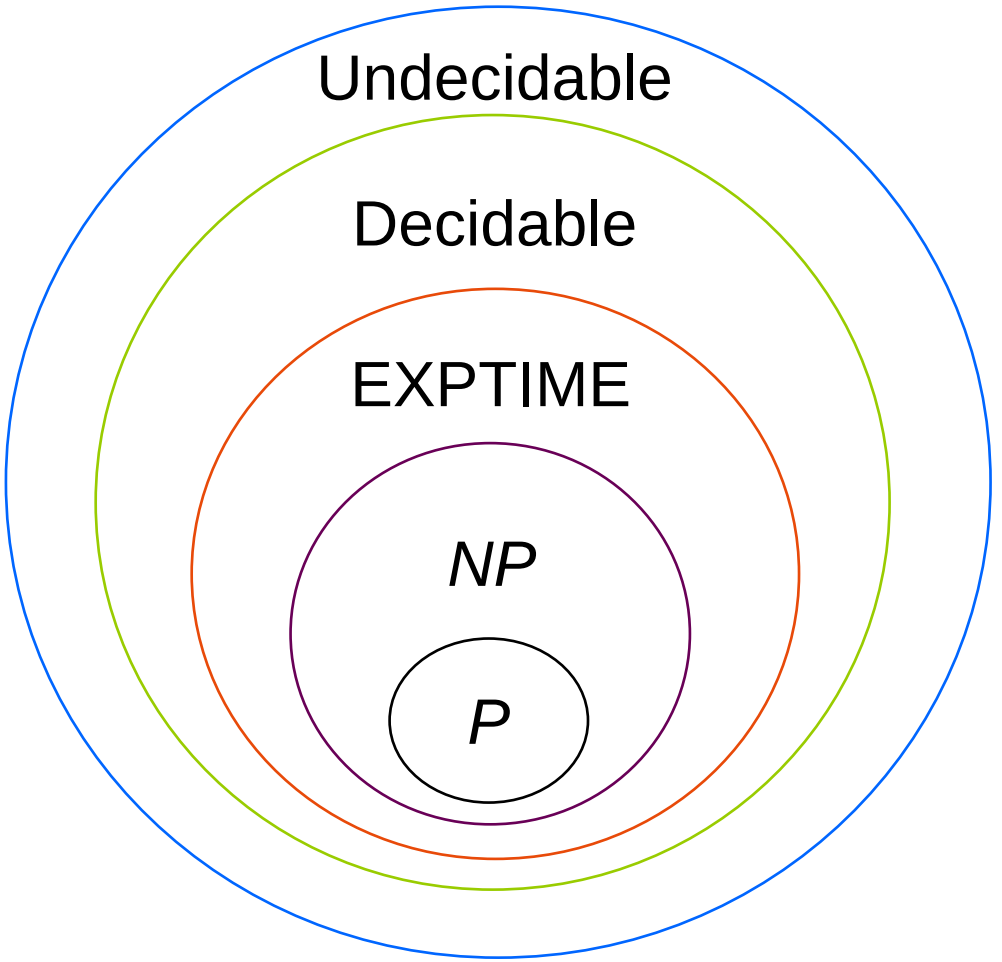
Motivation for this Talk

- Make it clear there are intriguing fundamental open scientific questions that are still lying at the heart of computer science.
- Positive answers to these questions will likely have a bigger impact on the world than even the Internet.
- This talk is an updated version of a discussion on the fundamental problems of computer science with Bill Gates during his visit to Columbia on October 12, 2005.

Bill Gates Roundtable with CS Faculty at Columbia University 10/13/05

QUESTION 1

- **How do we determine the difficulty of a problem?**



Time Complexity Hierarchy

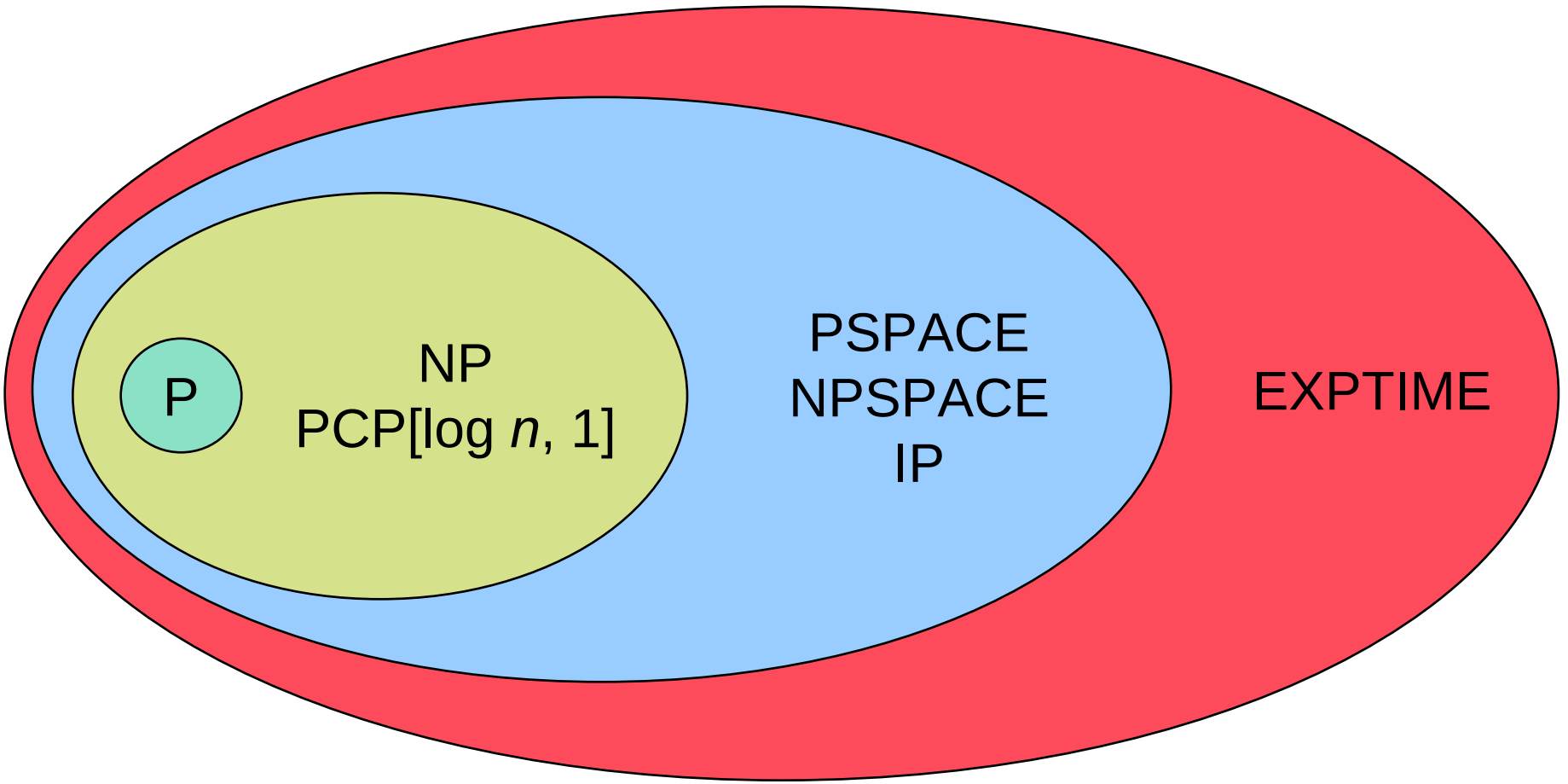
What We Know

- **There are undecidable problems (problems for which there are no algorithmic solutions).**
  (Alan Turing, 1936)

- **There are decidable problems that cannot be solved in exponential time.**
  (Follows from the time-hierarchy theorem, Hartmanis and Stearns, 1965)

Containment Relationships within EXPTIME

- **P ⊆ NP ⊆ PSPACE = NPSPACE = IP ⊆ EXPTIME**
- **P ≠ EXPTIME**



The Classes *P* and *NP*

- **A problem is in *P* if it can be solved in polynomial time by a deterministic Turing machine.**

  Example: Does a set of *n* positive and negative integers have a nonempty subset whose sum is positive?

  { –2, 7, –3, 14, –10, 15 }

- **A problem is in *NP* if it can be solved in polynomial time by a nondeterministic Turing machine.**

  Example: Does a set of *n* positive and negative integers have a nonempty subset whose sum is zero?

  { –2, 7, –3, 14, –10, 15 }

The *P* vs. *NP* Problem

- **Does *P* = *NP*?**
- **Informally: Are there any problems for which a computer can verify a given solution quickly but cannot find the solution quickly?**
- **Note: This is one of the seven Clay Mathematics Institute Millenium Prize Problems. The first person solving this problem will be awarded one million US dollars by the CMI.**

http://www.claymath.org/millennium/

Mainstream Media Fascination with the *P* vs. *NP* Problem

In the past two years the New York Times has published two articles about *P* vs. *NP*:

- ***Prizes Aside, the P-NP Puzzler has Consequences***
  (John Markov, October 7, 2009)

- ***Step 1: Post Elusive Proof. Step 2: Watch Fireworks***
  (John Markov, August 16, 2010)

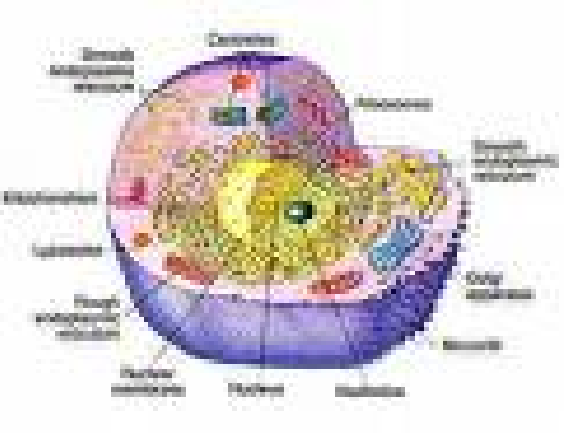
Another Open Problem: Integer Factoring

- **Problem: Given an *n*-bit integer, find all of its prime factors.**
- **Best-known deterministic algorithm has time complexity $\exp(O(n^{1/3}\log^{2/3} n))$.**
- **Open Problem: Can this problem be done in deterministic polynomial time?**

QUESTION 2

- **How do we model the behavior of complex systems that we would like to understand?**

Large distributed reactive systems such as the Internet or computing clouds



Human cell

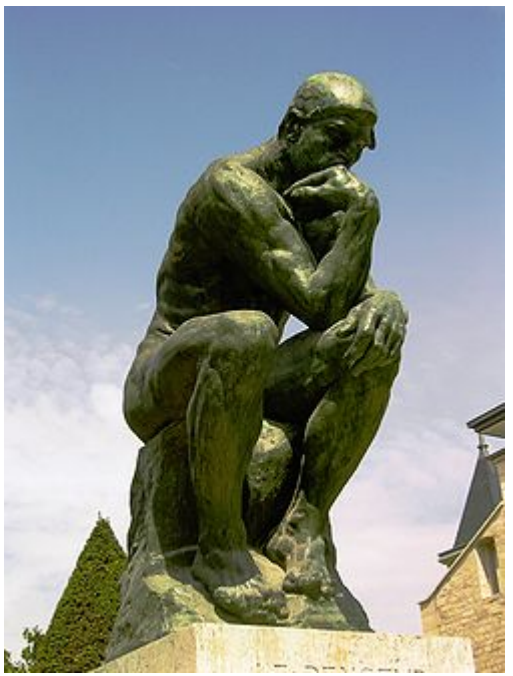
Open Question

- **Consider reactive distributed systems such as the Internet, cloud computing, and biological organisms that maintain an ongoing interaction with their environments.**

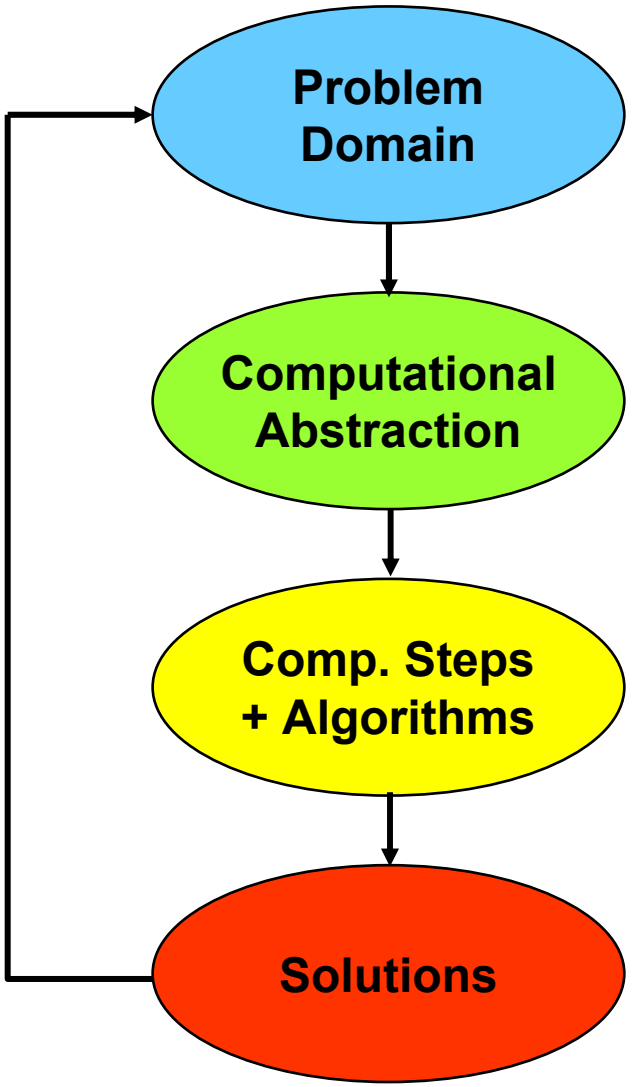
- **Is there a universal model of computation for such systems that is analogous to the Turing machine for sequential computation?**

Computational Thinking

The thought processes involved in formulating problems so their solutions can be represented as computational steps and algorithms.



Computational Thinking



The Four Postulates of Quantum Mechanics

1. The state of an isolated physical system can be described by a unit vector in a complex Hilbert space.
2. The evolution of a closed quantum system can be described by a unitary transformation.
3. Quantum measurements can be described by a collection { M_m } of measurement operators acting on the state space of the system being measured. The measurement operators satisfy the completeness equation

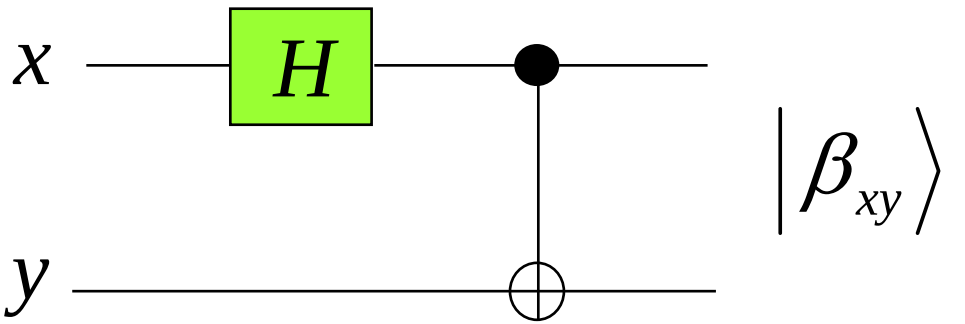
$$\sum_m M_m^\dagger M_m = I$$

4. The state space of a composite physical system is the tensor product of the state spaces of its composite systems.

M. Nielsen and I. Chuang
Quantum Computation and Quantum Information
Cambridge University Press, 2000

Computational Model for Quantum Computing: Quantum Circuits

Circuit to create Bell (Einstein-Podulsky-Rosen) states:



Circuit maps

$$|00\rangle \mapsto \frac{(|00\rangle + |11\rangle)}{\sqrt{2}}, |01\rangle \mapsto \frac{(|01\rangle + |10\rangle)}{\sqrt{2}}, |10\rangle \mapsto \frac{(|00\rangle - |11\rangle)}{\sqrt{2}}, |11\rangle \mapsto \frac{(|01\rangle - |10\rangle)}{\sqrt{2}}$$

Each output is an entangled state, one that cannot be written as a product of two single-qubit states.

Einstein: "Spooky action at a distance."

Shor's Integer Factorization Algorithm

Problem: Given a composite *n*-bit integer, find a nontrivial factor.

Best-known deterministic algorithm on a classical computer has time complexity $\exp(O(n^{1/3}\log^{2/3} n))$.

A quantum computer can solve this problem in $O(n^3)$ operations.

Peter Shor

Algorithms for Quantum Computation: Discrete Logarithms and Factoring
***Proc. 35th Annual Symposium on Foundations of Computer Science*, 1994, pp. 124-134**

Integer Factorization: Estimated Times

Classical: number field sieve

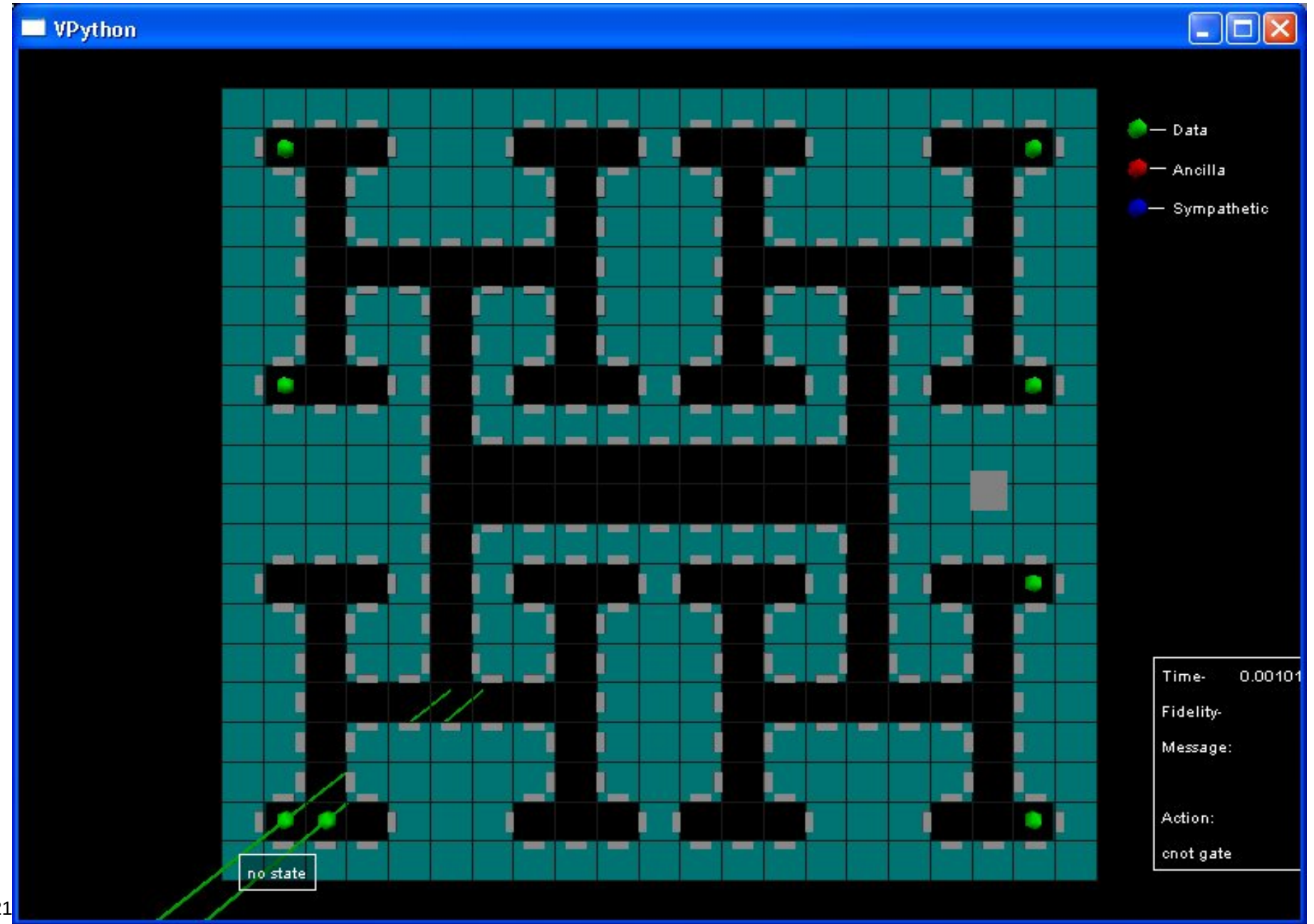
- Time complexity: $\exp(O(n^{1/3} \log^{2/3} n))$
- Time for 512-bit number: 8400 MIPS years
- Time for 1024-bit number: 1.6 billion times longer

Quantum: Shor's algorithm

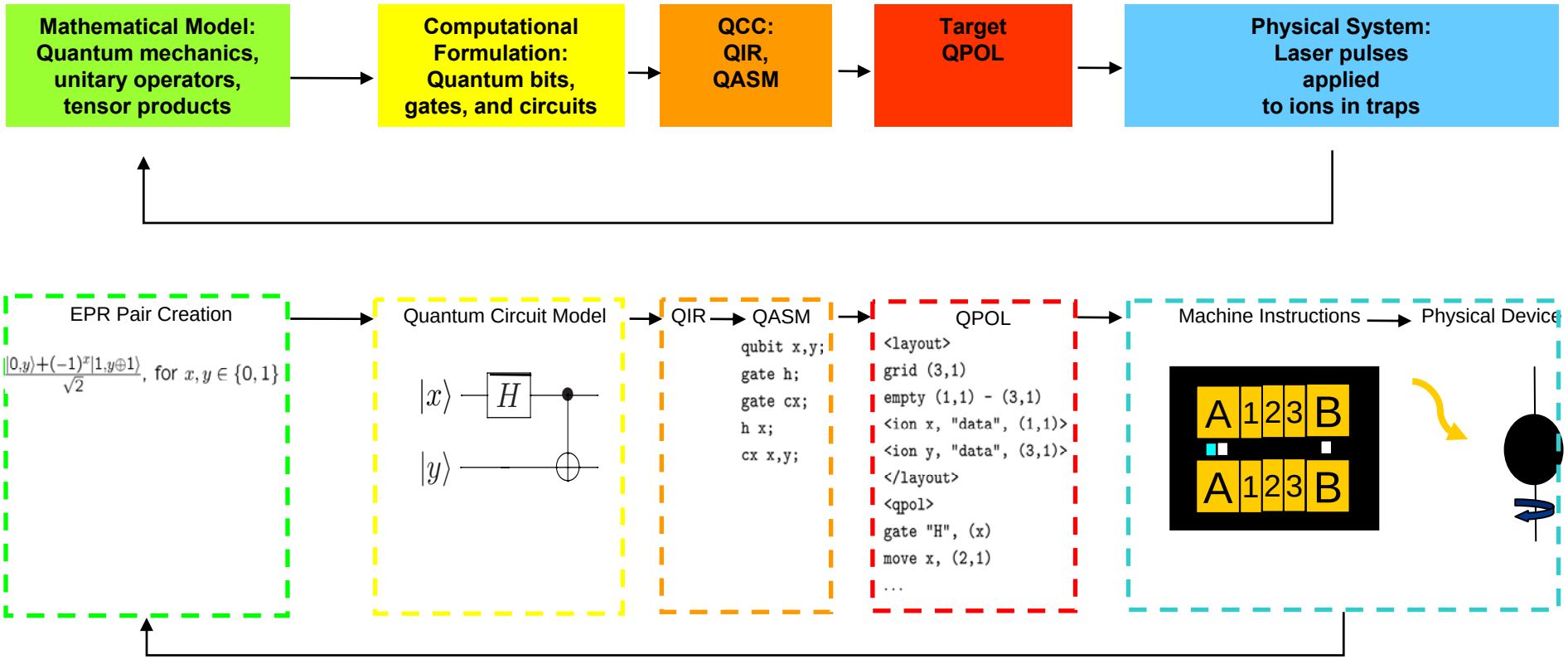
- Time complexity: $O(n^3)$
- Time for 512-bit number: 3.5 hours
- Time for 1024-bit number: 31 hours
  (assuming a 1 GHz quantum device)

M. Oskin, F. Chong, I. Chuang
A Practical Architecture for Reliable Quantum Computers
IEEE Computer, 2002, pp. 79-87

Ion Trap Quantum Computer



Quantum Computer Compiler



K. Svore, A. Aho, A. Cross, I. Chuang, I. Markov

A Layered Software Architecture for Quantum Computing Design Tools

IEEE Computer, 2006, vol. 39, no. 1, pp.74-83

Open Question Reprised

- **Consider reactive distributed systems such as the Internet, cloud computing, and biological organisms that maintain an ongoing interaction with their environments.**

- **Is there a universal model of computation for such systems that is analogous to the Turing machine for sequential computation?**

QUESTION 3

- **Can we build secure networked systems?**

The global Internet

Demand for System Security: Protection from Malware

- **Internet malware**
  - worms, viruses, spyware and Internet-cracking tools
  - worms override program control to execute malcode
- **Internet worms**
  - Morris '88, Code Red II '01, Nimda '01, Slapper '02, Blaster '03, MS-SQL Slammer '03, Sasser '04, Conficker '08
  - automatic propagation
- **Internet crackers**
  - "**j00** *got* **h4x0r3d**!!"
- **After breaking in, malware will**
  - create backdoors, install root kits (conceal malcode existence), join a botnet, generate spam, damage equipment

Worms, viruses prove costly

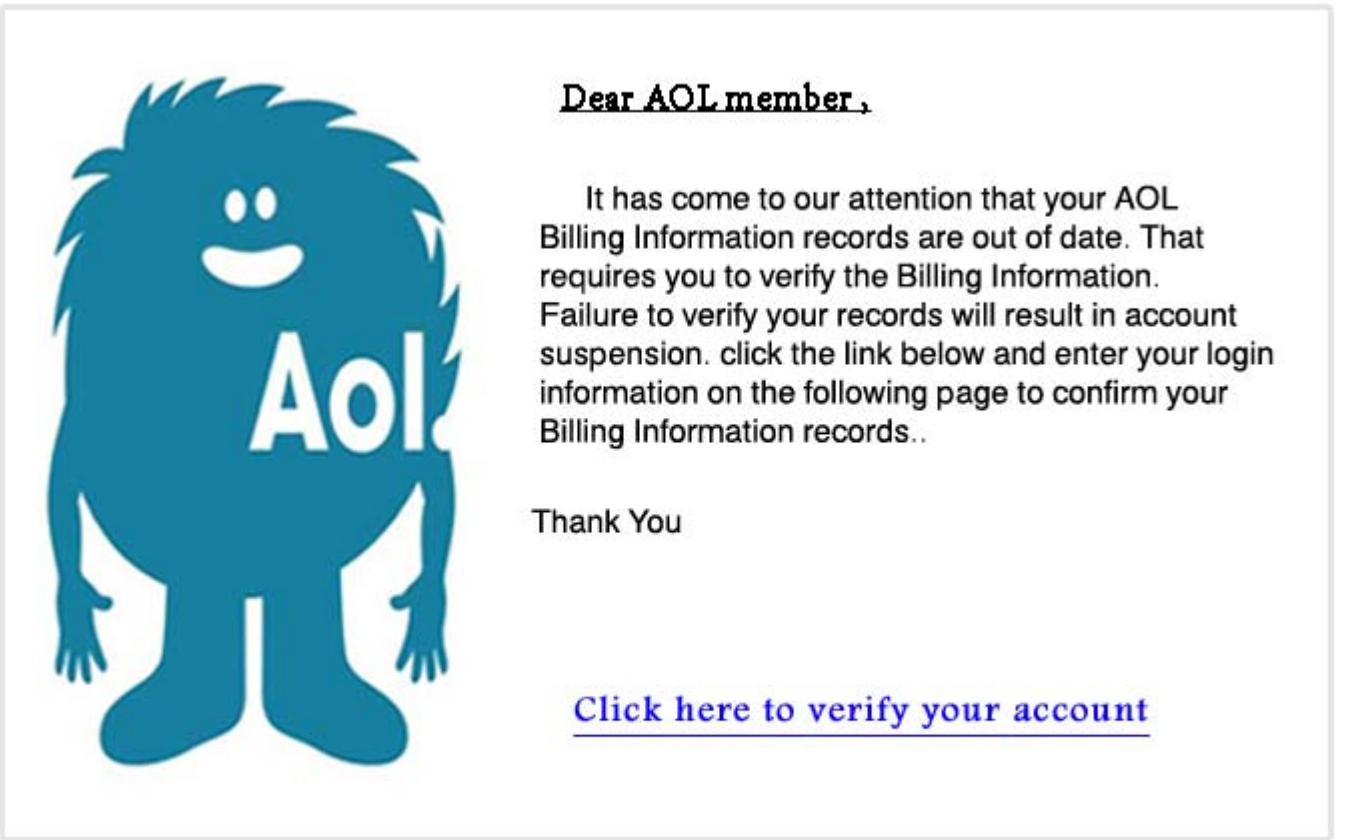
The estimated cleanup and lost productivity costs of worms and viruses add up:

Year	Virus/worm	Estimated damage
1999	Melissa virus	$80 million
2000	Love Bug virus	$10 billion
2001	Code Red I and II worms	$2.6 billion
2001	Nimda virus	$590 million to $2 billion
2002	Klez worm	$9 billion
2003	Slammer worm	$1 billion

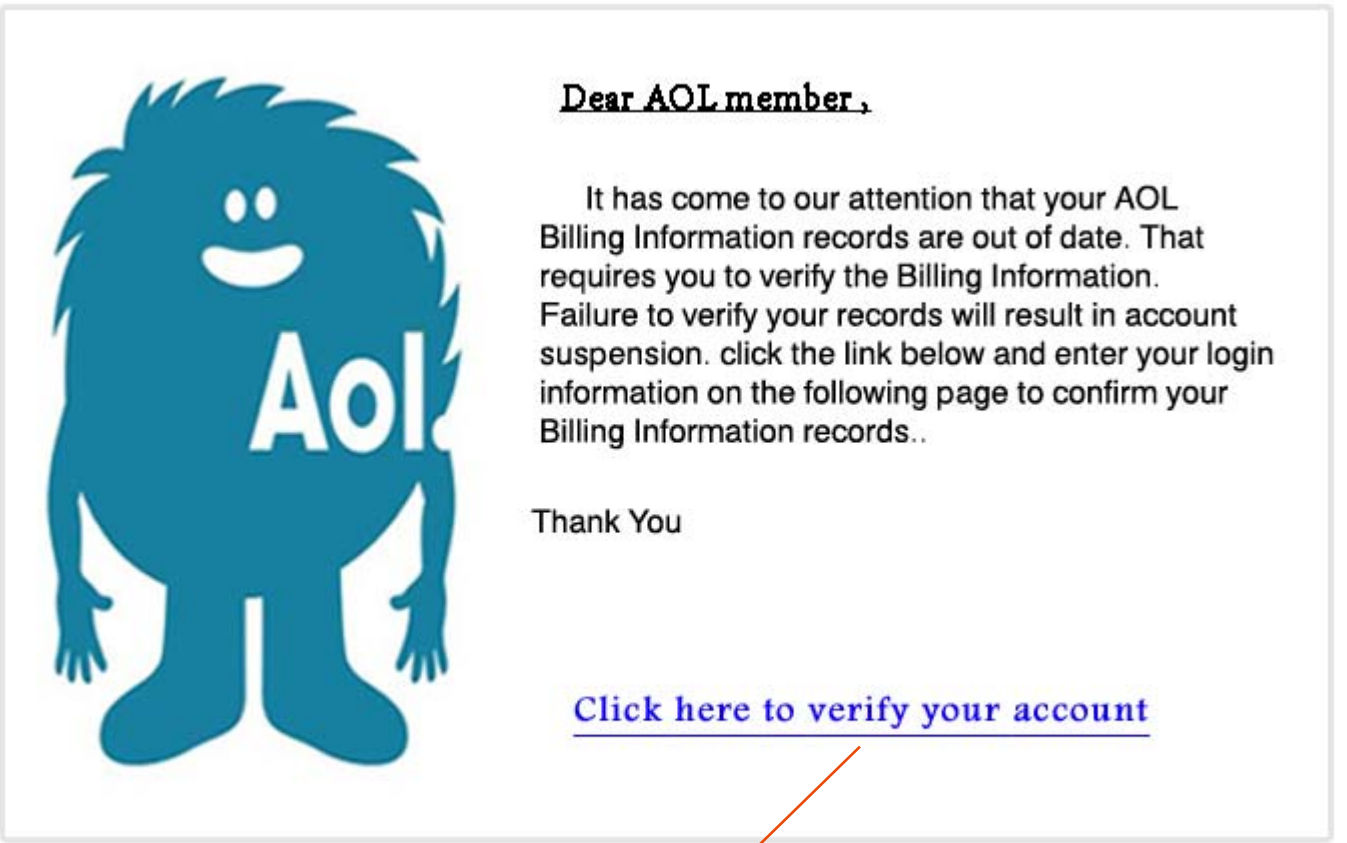
Source: USA TODAY research

Phishing Attacks: Spear-phishing and whaling

Selective phishing attacks that use relevant contextual information to trick specific victims



Phishing Attacks: Spear-phishing



http://snss.aolbillingcenter.com.ruangmobil.com/Login/Aol/Update-Billing/update_form!!!

Cyberwarfare: Stuxnet

- **On 17 June 2010, VirusBlokAda [an antivirus-software company in Minsk] issued a worldwide alert that set off an international race to track down what came to be known as Stuxnet: the most sophisticated computer malware yet found and the harbinger of a new generation of cyberthreats.**

- **Unlike conventional malware, which does its damage only in the virtual world of computers and networks, Stuxnet would turn out to target the software that controls pumps, valves, generators and other industrial machines.**

S. Weinberger
Is this the start of cyberwarfare?
Nature, vol. 474, 9 June 2011, pp. 142-145

Unusual Properties of the Stuxnet Worm

- **Staggering size: 15,000 lines of code, representing an estimated 10,000 person hours in software development**
- **Includes a programmable logic controller (PLC) rootkit**
- **Targets Siemens Supervisory Control and Data Acquisition (SCADA) systems used to control specific industrial processes**
- **Used sophisticated combination of attacks: four zero-day Windows exploits, root-mode and kernel-mode rootkits, and device drivers that were digitally signed with two stolen private keys**

N. Falliere, L. O Murchu, E. Chien
W32.Stuxnet Dossier
Symantec Security Response, February, 2011

Open Questions

- How should we define security (and privacy)?
- What are acceptable levels of security (and privacy) for computer systems?
- How can we achieve these levels?

QUESTION 4

- **Is there a scientific basis for making reliable software?**

How Can We Make Reliable Software?

- **Communication:** Shannon [1948] used error detecting and correcting codes for reliable communication over noisy channels
- **Hardware:** von Neumann [1956] used redundancy to create reliable systems from unreliable components
- **Software:** Is there a similar scientific basis for making reliable software?

Volume of Software and Defects

- **World uses hundreds of billions of lines of software**
  - **5 million programmers worldwide**
  - **average programmer generates 5,000 new lines of code annually**
  - **embedded base: many hundreds of billions of lines of software**
- **Number of embedded defects**
  - **defect densities: 10 to 10,000 defects/million lines of code**
  - **total number of defects in embedded base: 5×10^6 to 50×10^9**

A. V. Aho
Software and the Future of Programming Languages
Science, vol. 303, 27 February 2004, pp. 1331-1333

IEEE Spectrum Software Hall of Shame

Year	Company	Costs in US $
2004	UK Inland Revenue	Software errors contribute to $3.45 billion tax-credit overpayment
2004	J Sainsbury PLC [UK]	Supply chain management system abandoned after deployment costing $527M
2002	CIGNA Corp	Problems with CRM system contribute to $445M loss
1997	U. S. Internal Revenue Service	Tax modernization effort cancelled after $4 billion is spent
1994	U. S. Federal Aviation Administration	Advanced Automation System canceled after $2.6 billion is spent

R. N. Charette, *Why Software Fails,* IEEE Spectrum, September 2005.

Columbia Programming Languages and Compilers Course: Goals I

- Teach the fundamental principles of
  - Programming languages
  - Compilers

Columbia Compilers Course: Goals II

- **Teach computational thinking**
  - **How to create and use abstractions for software design and development**

Columbia Compilers Course: Goals III

- Teach robust software engineering skills
  - Project management
  - Teamwork
  - Communication

Project Specification

- **Form a team of five to create and implement an innovative little language of your own design**

Project Process, Timeline and Deliverables

Week	Deliverable
2	Form team of five, assign responsibilities
4	Write language whitepaper (vision statement)
4-8	Prototype language
8	Write language tutorial and reference manual
8-15	Implement and test compiler incrementally
14	Present language to class
15	Hand in final project report
15	Demo compiler to teaching staff

Final Project Report I

1. Introduction (language whitepaper) [team]
2. Language tutorial [team]
3. Language reference manual [team]
4. Project plan [project manager]
   - Process used
   - Roles and responsibilities
   - Implementation style sheet
   - Timeline
   - Project log
5. Translator architecture [system architect]
   - Architectural block diagram of translator
   - Interfaces
   - Responsibilities

Final Project Report II

6. Development environment [system integrator]
   - Software development environment used
   - Implementation languages and tools used
   - Libraries used
7. Test plan and test suites [V&V person]
   - Test methodology used
   - Show programs used to test your translator
8. Conclusions [team + each team member]
   - Lessons learned as a team
   - Lessons learned by each team member
   - Advice for future teams and the instructor
9. Appendix [team members]
   - Source code listing with identification of who wrote which module

Ingredients for Making Reliable Software

- Good people
- Modeling and prototyping
- Type-safe programming languages
- Sound software engineering practices
- Daily one-click builds
- Tools incorporated into the software development lifecycle

But the Open Problem Remains

Is there a scientific basis for making reliable software?

QUESTION 5

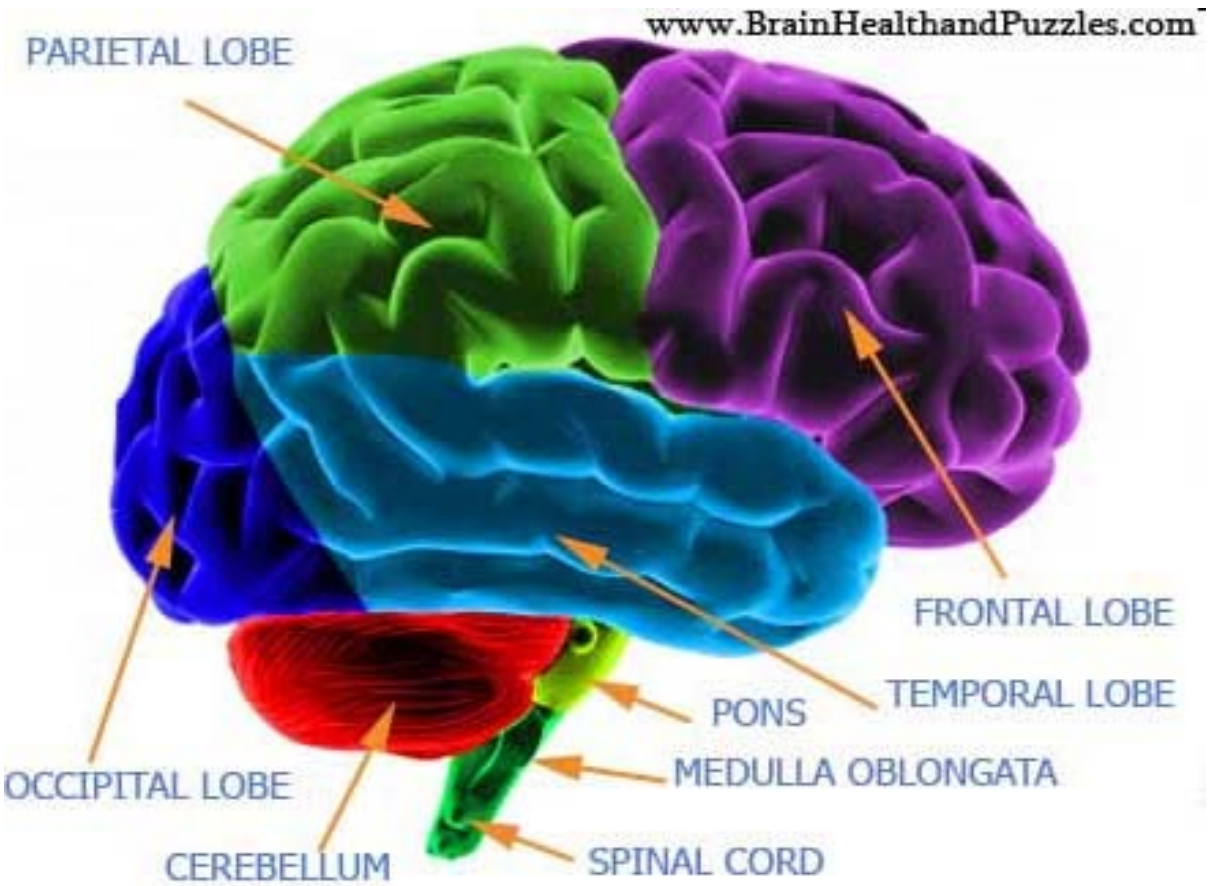
- **Can we construct computer programs that exhibit human-like attributes such as consciousness or intelligence?**

Cogito, ergo sum.

Can Machines Think?

- In his 1950 paper, *Computing Machinery and Intelligence*, Alan Turing proposed the quintessential question, "Can machines think?"
- He then rephrased the question with "Are there imaginable digital computers which would do well in the imitation game?"
- Loebner prize, initiated in 1990, is awarded annually to the chatterbot considered most human-like. The human judges have been deceived several times into thinking programs were humans.
- Eliza, a simple pattern-matching program written in 1964 by Joseph Weizenbaum, has fooled many people into thinking it is a Rogerian psychotherapist.

Open Question: Is thinking just computation?

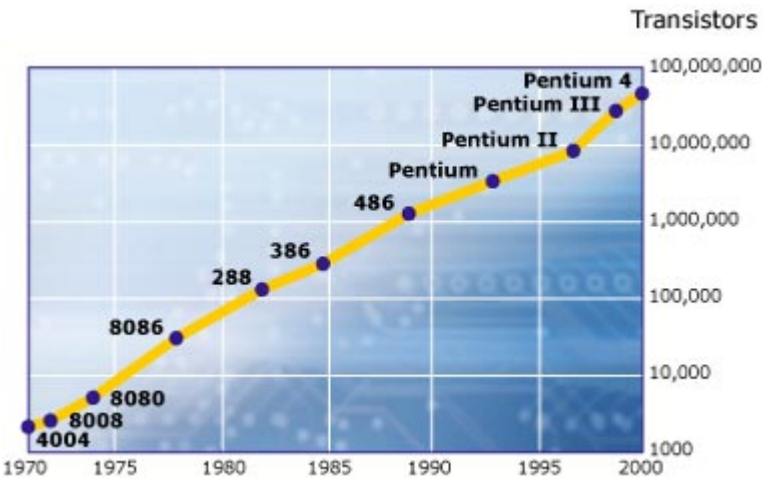


Bill Gates's Responses

1. How do we determine the difficulty of a problem?
2. How do we model the behavior of complex systems that we would like to understand?
3. Can we build secure networked systems?
4. Is there a scientific basis for making reliable software?
5. Can we construct computer programs that exhibit human-like attributes such as consciousness or intelligence?

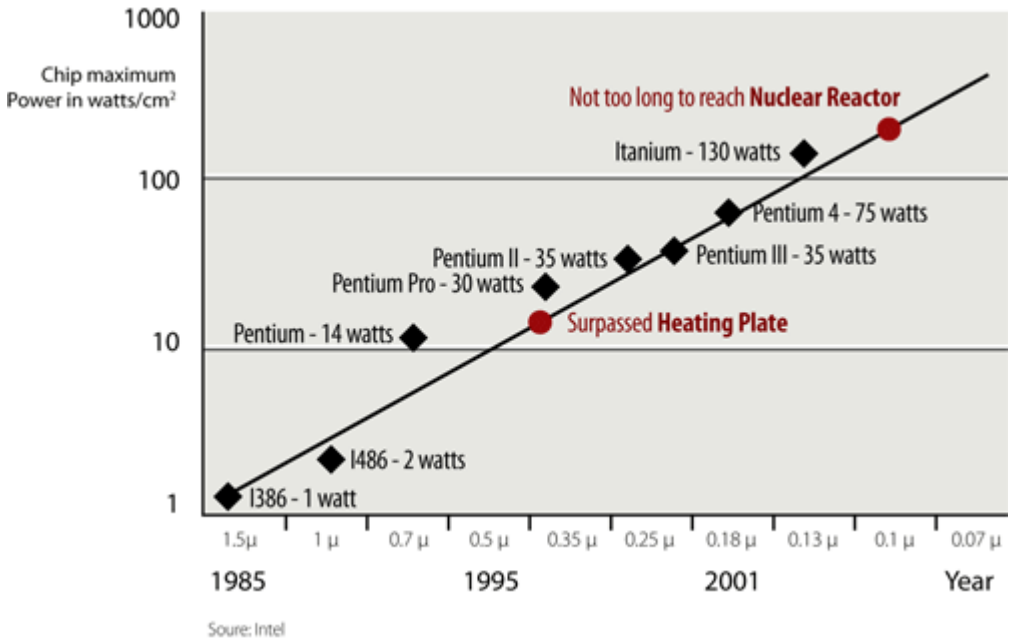
Bill Gates's Question

- **What is the best way to extend Moore's Law?**



Moore's Law for Number of Transistors

Power Consumption



Moore's Law for Power Consumption

QUESTION 6

- **What is the best way to extend Moore's Law?**

Summary

- There are intriguing fundamental open scientific questions that are still lying at the heart of computer science.

- Positive answers to these questions will likely have a bigger impact on the world than even the Internet.

Al Aho

aho@cs.columbia.edu

Thanks for Listening!

The Quintessential Questions of Computer Science

Bell Labs

Murray Hill, NJ

June 22, 2011