The Discovery of the Factoring Algorithm

ICTP, Trieste

Peter Shor
M.I.T.
Cambridge, MA

Outline of Talk

1. The first few slides of my 1990s factoring talk (somewhat updated).

2. How I came to discover the factoring algorithm.

3. Some aftereffects of the discovery.

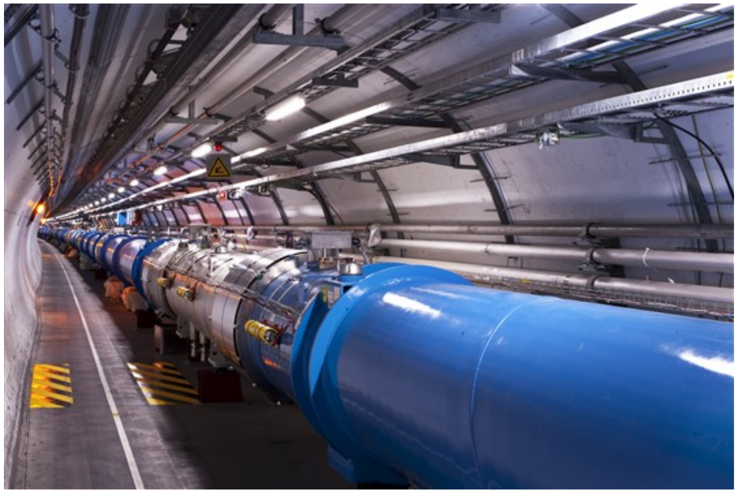
What is the difference between a computer and a physics experiment?

One answer:

A physics experiment is a big, custom-built, finicky, piece of apparatus.

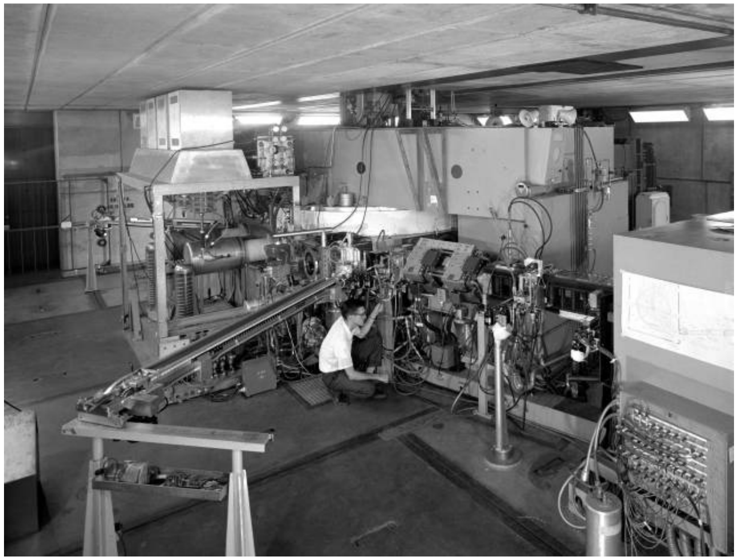
A computer is a little box that fits in your briefcase.

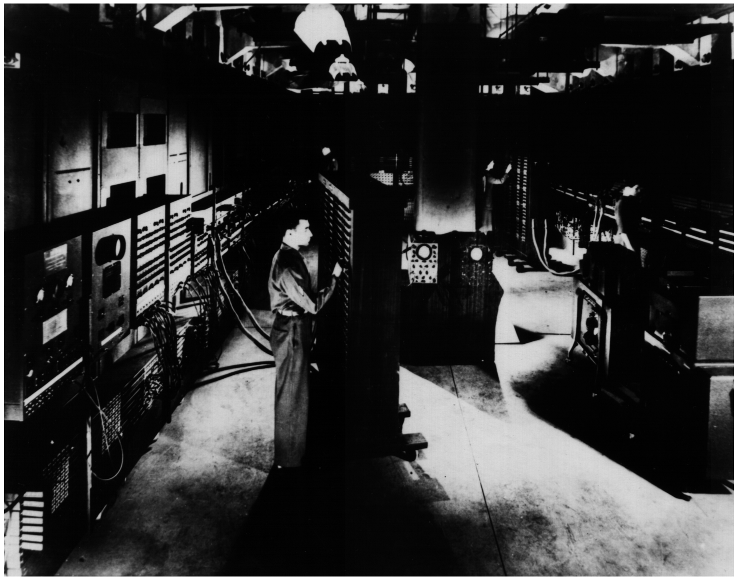
Physics experiment



Computer

Physics experiment



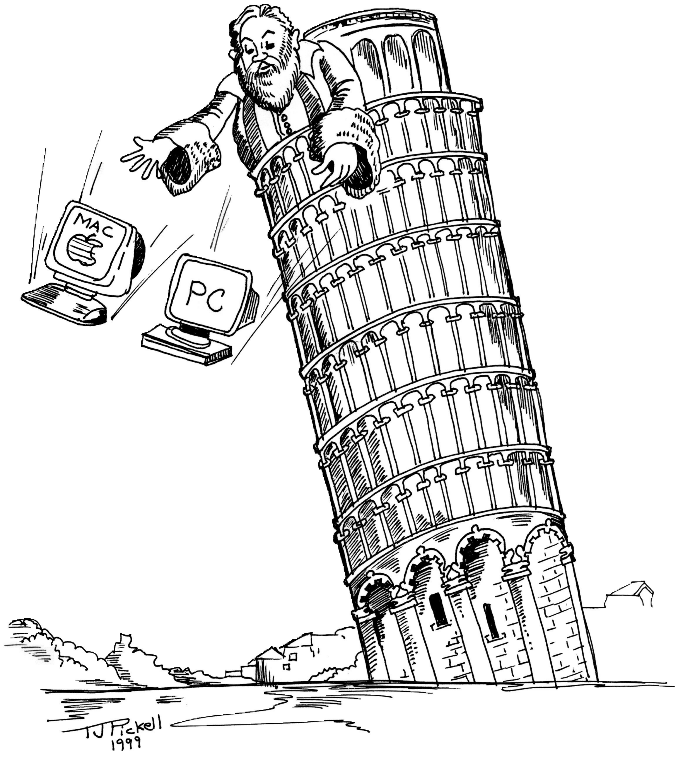


Computer

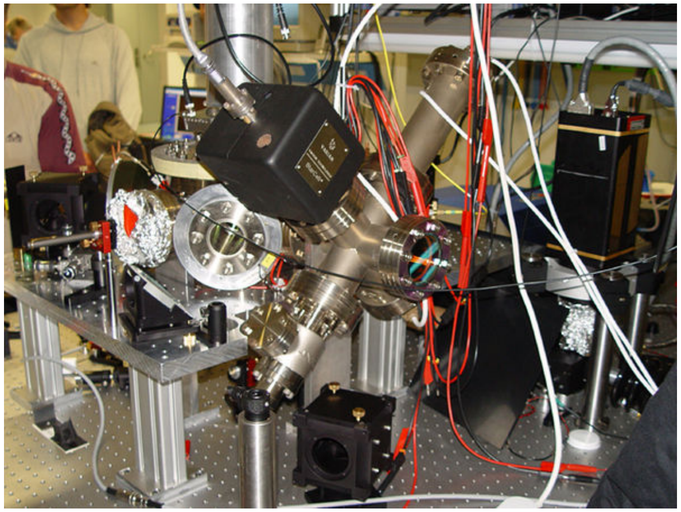
A second answer:

A computer answers mathematical questions.
A physics experiment answers physical questions.

Using computers to test whether two bodies fall at the same rate



Using a physics experiment to solve the factoring problem $15 = ? \cdot ?$ (Rainer Blatt's group, Innsbrück, Austria).



A third answer:

You don't need to build a new computer for each mathematical question you want answered.

This means that you can mass-produce computers, while it's hard to mass-produce physics experiments. (Although you can mass-produce components for them.)

This is related to a fundamental fact about computation:

Universality of computation

Church-Turing thesis:

A Turing machine can perform any computation that any device can perform. (Turing, Church, ca. 1936).

Inadequacy of the Church-Turing thesis in practice.

With the development of practical computers, the distinction between uncomputable and computable become much too coarse.

To be practical, a program must compute a function in a reasonable amount of time (in years, at the longest).

Theoretical computer scientists came up with the idea of “efficient” means polynomial time as a workable compromise between theory and practice.

Universality of computation II.

Various computer scientists proposed

Quantitative Church's Thesis (Cobham)

A Turing machine can perform *efficiently* any computation that any (physical) device can perform efficiently.

(Various theoretical computer scientists, 1960's).

If quantum computers can be built, this would imply this "folk thesis" is not true.

Misconceptions about Quantum Computers

False: Quantum computers would be able to speed up all computations.

Quantum computers are not just faster versions of classical computers.

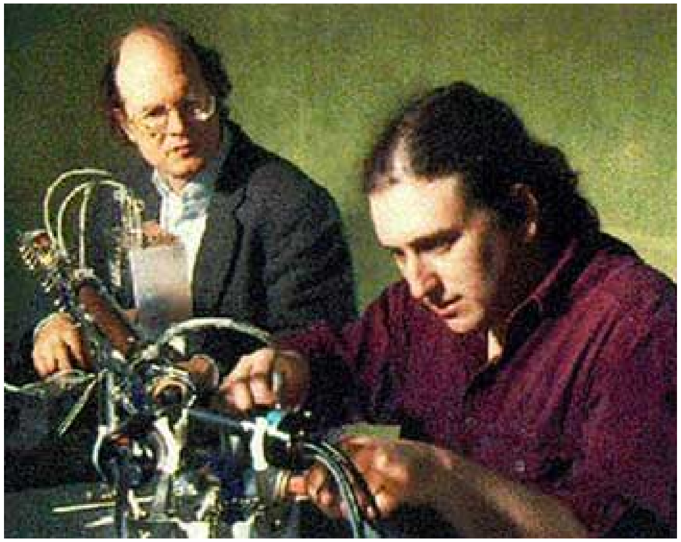
They would speed up some problems by large factors and other problems not at all.

The fact that this misconception is so widespread shows that the public has absorbed the Quantitative Church's Thesis.

A single step on a quantum computer is almost certain to take longer than a single step on a classical computer. Quantum computers speed up computations by drastically reducing the number of steps needed.

Part II: What led up to the discovery.

My first exposure to quantum computing was when I heard a talk Charlie Bennett gave at Bell Labs about quantum key distribution.



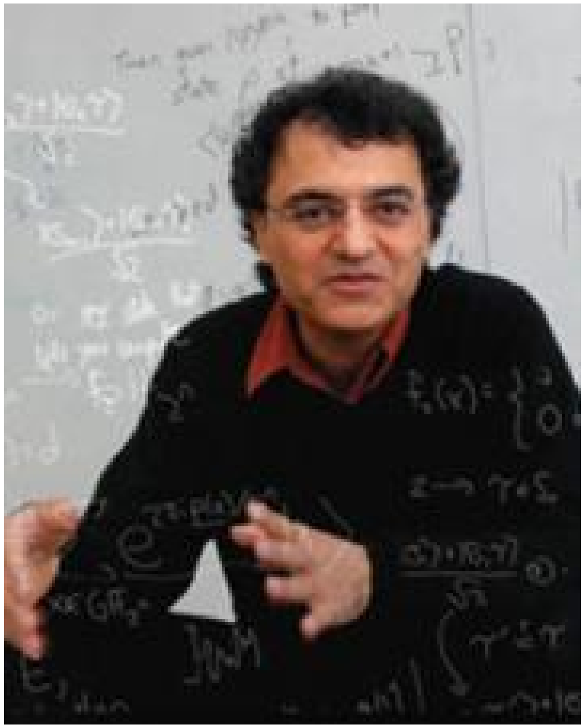
David Deutsch's papers

I looked at David Deutsch's papers on quantum computing.

I wasn't convinced by them that Deutsch had a rigorous mathematical description of quantum computing (this was actually my fault) or that it was at all useful.

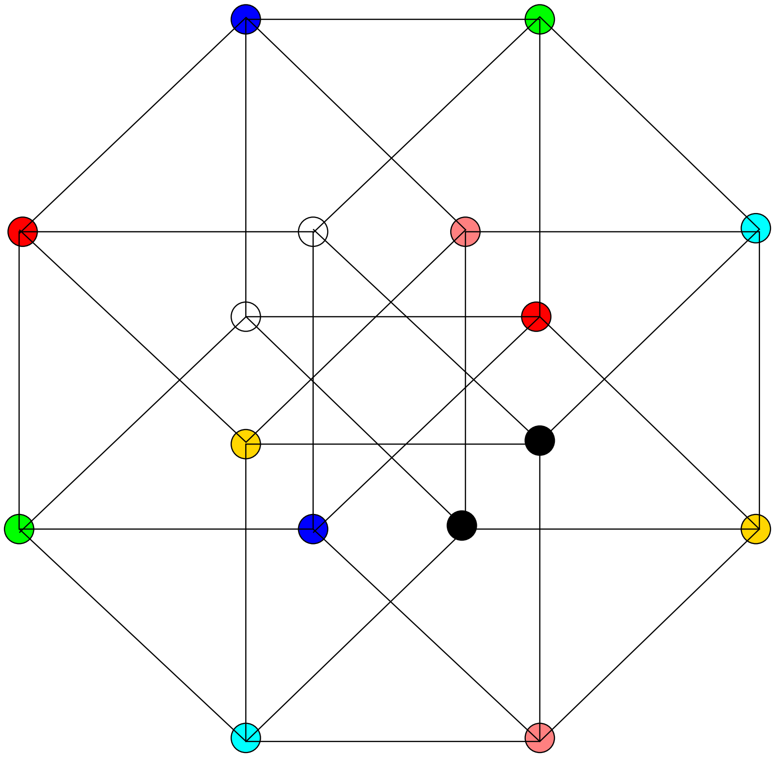
Umesh Vazirani

Then Umesh Vazirani gave a talk at Bell Labs about his paper "Quantum Complexity Theory" with Ethan Bernstein.
This started me thinking seriously about quantum computing.



Simon's Algorithm

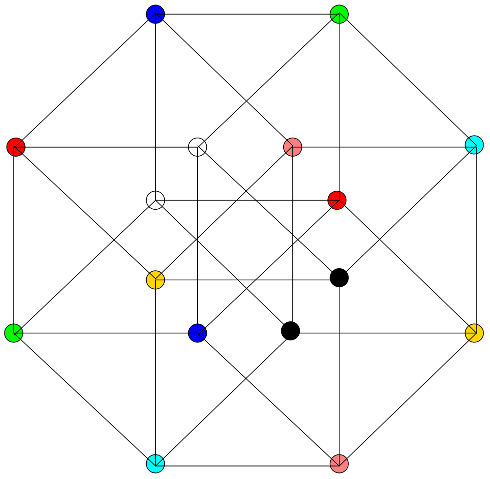
Next, I was on a conference program committee. Dan Simon had submitted the paper containing algorithm to this committee, and I saw it. (We rejected it!)



Simon's Algorithm → Discrete Log

Simon's algorithm contained a lot of the ingredients I needed to discover the factoring algorithm.

It had periodicity (mod 2).
It had the Fourier transform (over Z_2^n).

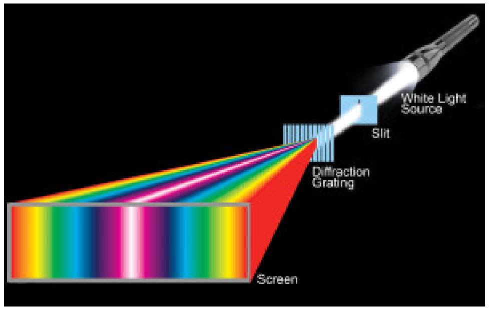


I knew that the discrete log problem could be solved by using periodicity, and after several months of thinking about it (part time), I discovered the discrete log algorithm.

How does the factoring algorithm work?

The quantum Fourier transform can function as a computational interferometer.

A diffraction grating uses interference to separate the different wavelengths of light, so each wavelength ends up at a different spot.



The quantum Fourier transform uses interference to separate the possible periods of a periodic function, so each different period results in a different output.

Part 3: After the discovery

I gave a talk at Bell Labs about the algorithm for discrete log on a Tuesday in April, 1994, Henry Landau's seminar.

That weekend, Umesh Vazirani, very excited, called me at home and said "I hear you can factor on a quantum computer. Tell me how it works." I explained the algorithm to him.

Note that the rumor switched from discrete log to factoring on its way to him. Luckily, I'd solved the factoring problem in the intervening 5 days.

Word of the results spread.

Shortly after, I got a phone call from *the Economist.*

In the next few months, I got tons of emails asking for the paper (it wasn't written yet).

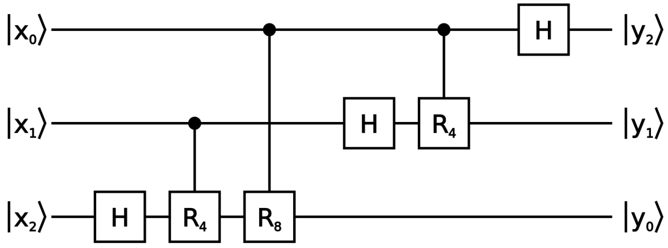
There were lots of talks about the algorithm at conferences. In May, I gave one at ANTS in Cornell. In June, Umesh, gave one at the Santa Fe Institute. In August, I gave one at NIST, and Artur Ekert gave one at ICAP. In October, I gave one at the Villa Gualino in Torino.

Dan Simon's paper and my paper were both accepted at the FOCS conference that November.

The quantum circut model.

I started describing quantum computers as quantum Turing machines, which was what the Bernstein-Vazirani paper talked about.

Talking to physicists made me realize that quantum Turing machines would be almost impossible to realize, and that this made them very difficult to explain to physicists. So I switched to the quantum circuit model, which I believe was first described by David Deutsch.



One objection

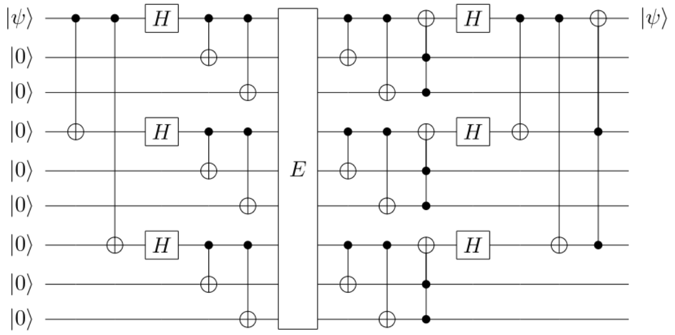
One objection to the factoring result was that if you needed to do 10^9 steps on a quantum computer, each gate had to be accurate to about one part in 10^9. This was completely out of reach experimentally.

And quantum computers couldn't be made fault tolerant ... you couldn't use redundancy because of the no-cloning theorem, and if you measured to see whether there was an error, the Heisenberg Uncertainty Principle mean that you would disturb the quantum state and destroy the computation.

Objection resolved (to some extent)

In fact, I (along with others) showed that quantum error correcting codes existed and quantum computers can be made fault tolerant.

You arrange the codes that the likely errors are orthogonal to the encoded state, and then you can measure the errors without disturbing the encoded state.



So you only need gates accurate to around one part in 10^4. This is still very difficult experimentally, but not hopelessly out of reach.