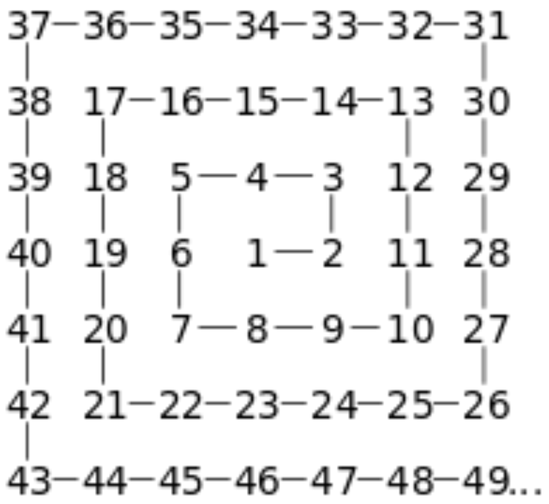
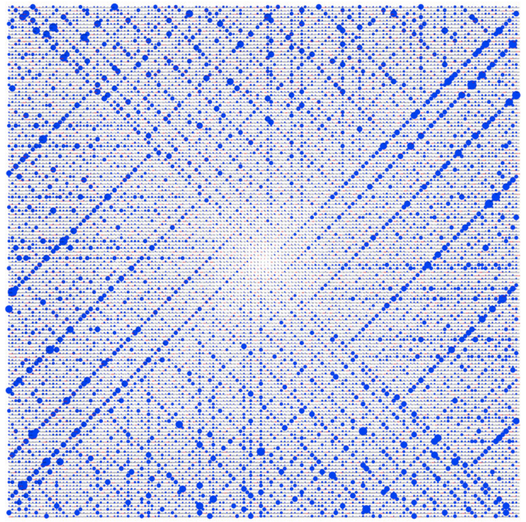
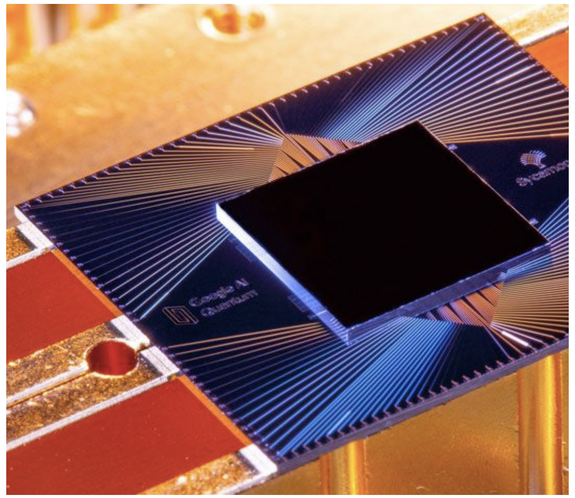
From Prime Numbers to Quantum Computing

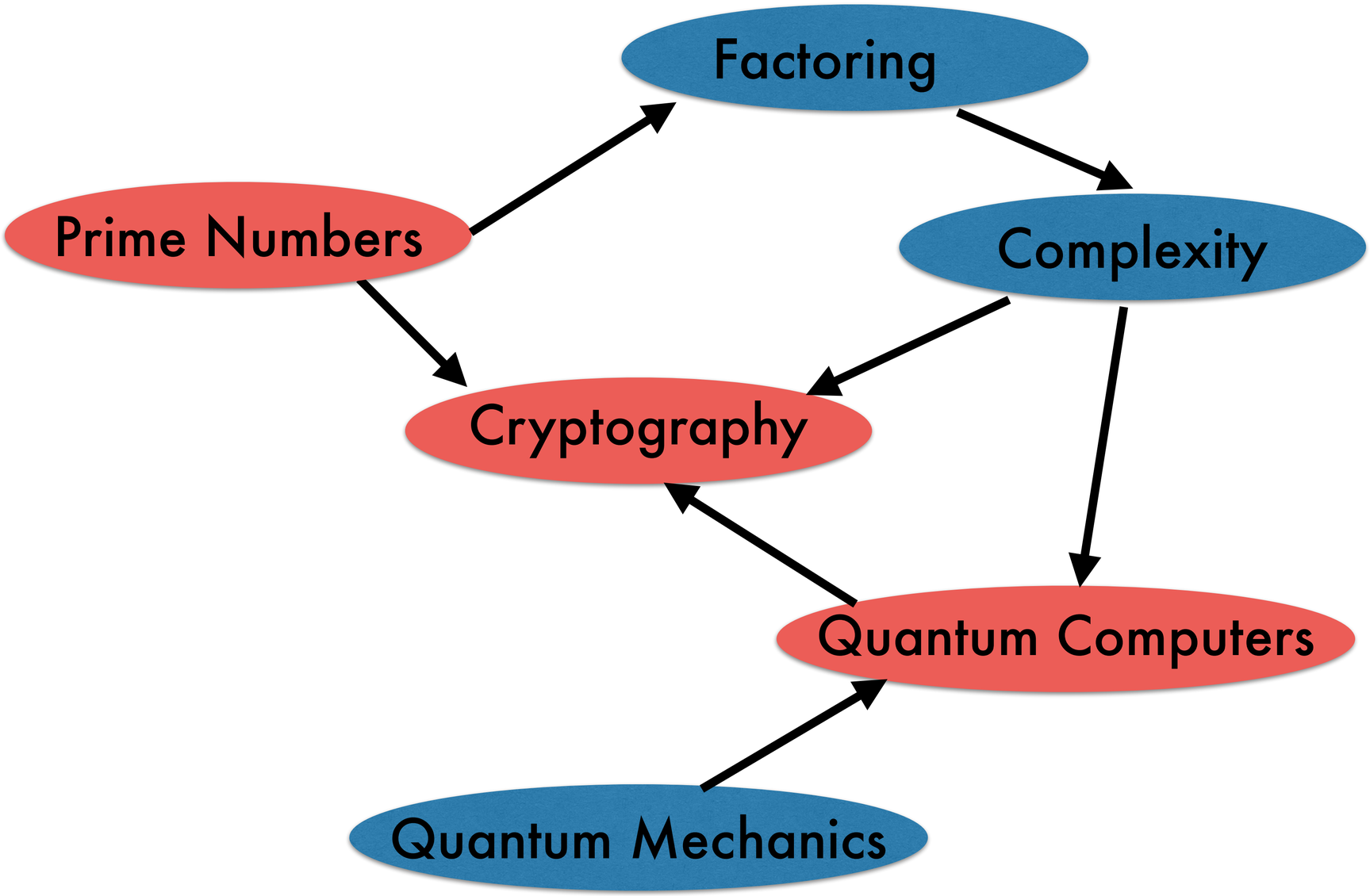
Luca Doria (doria@uni-mainz.de)
PRISMA+ Cluster of Excellence and Institute for Nuclear Physics
Johannes Gutenberg University Mainz







Lecture Summary



Prime Numbers

Prime Numbers (Primzahlen)

Prime number: divisible (reminder = 0) only by 1 and itself.

Examples: (1), 2, 3, 5, 7, 11, 13, 17, 19, 23, 29,.....

Prime Numbers (Primzahlen)

Prime number: divisible (reminder = 0) only by 1 and itself.
Examples: (1), 2, 3, 5, 7, 11, 13, 17, 19, 23, 29,.....

Every number can be factorized in primes: 105 = 3*5*7

Prime Numbers (Primzahlen)

Prime number: divisible (reminder = 0) only by 1 and itself.
Examples: (1), 2, 3, 5, 7, 11, 13, 17, 19, 23, 29,.....

Every number can be factorized in primes: 105 = 3*5*7

Fundamental theorem: the factorization is unique

Prime Numbers (Primzahlen)

Prime number: divisible (reminder = 0) only by 1 and itself.
Examples: (1), 2, 3, 5, 7, 11, 13, 17, 19, 23, 29,.....

Every number can be factorized in primes: 105 = 3*5*7

Fundamental theorem: the factorization is unique

How many are the prime numbers?

Prime Numbers (Primzahlen)

Prime number: divisible (reminder = 0) only by 1 and itself.
Examples: (1), 2, 3, 5, 7, 11, 13, 17, 19, 23, 29,.....

Every number can be factorized in primes: 105 = 3*5*7

Fundamental theorem: the factorization is unique

How many are the prime numbers?
Euclid's Theorem: the prime numbers are infinite! Largest known: $2^{82.589.933}$-1

Prime Numbers (Primzahlen)

Prime number: divisible (reminder = 0) only by 1 and itself.
Examples: (1), 2, 3, 5, 7, 11, 13, 17, 19, 23, 29,.....

Every number can be factorized in primes: 105 = 3*5*7

Fundamental theorem: the factorization is unique

How many are the prime numbers?
Euclid's Theorem: the prime numbers are infinite! Largest known: $2^{82.589.933}$-1

Goldbach Conjecture: every even integer > 2 can be expressed as the sum of two primes (4=2+2, 6=3+3, 8=5+3, ...).

Prime Numbers (Primzahlen)

Prime number: divisible (reminder = 0) only by 1 and itself.
Examples: (1), 2, 3, 5, 7, 11, 13, 17, 19, 23, 29,.....

Every number can be factorized in primes: 105 = 3*5*7

Fundamental theorem: the factorization is unique

How many are the prime numbers?
Euclid's Theorem: the prime numbers are infinite! Largest known: $2^{82.589.933}$-1

Goldbach Conjecture: every even integer > 2 can be expressed as the sum of two primes (4=2+2, 6=3+3, 8=5+3, ...).
Proof ?

Prime Numbers (Primzahlen)

Prime Numbers (Primzahlen)

Primes are infinite! Can we say (at least!) something about their asymptotic behaviour?

Prime Numbers (Primzahlen)

Primes are infinite! Can we say (at least!) something about their asymptotic behaviour?

Prime Number Theorem $p_n \sim n \log n$

Prime Numbers (Primzahlen)

Primes are infinite! Can we say (at least!) something about their asymptotic behaviour?

Prime Number Theorem $p_n \sim n \log n$

Asymptotic distribution law: $\pi(x) \sim \dfrac{x}{\log x}$

Prime Numbers (Primzahlen)

Primes are infinite! Can we say (at least!) something about their asymptotic behaviour?

Prime Number Theorem $p_n \sim n \log n$

Asymptotic distribution law: $\pi(x) \sim \frac{x}{\log x}$

Sum of the inverses: $\sum_{primes\ n} \frac{1}{n} = \frac{1}{2} + \frac{1}{3} + \ldots =$

Prime Numbers (Primzahlen)

Primes are infinite! Can we say (at least!) something about their asymptotic behaviour?

Prime Number Theorem $p_n \sim n \log n$

Asymptotic distribution law: $\pi(x) \sim \frac{x}{\log x}$

Sum of the inverses: $\sum_{primes\ n} \frac{1}{n} = \frac{1}{2} + \frac{1}{3} + \ldots = \infty$

Prime Numbers (Primzahlen)

Primes are infinite! Can we say (at least!) something about their asymptotic behaviour?

Prime Number Theorem $p_n \sim n \log n$

Asymptotic distribution law: $\pi(x) \sim \frac{x}{\log x}$

Sum of the inverses: $\sum_{primes\ n} \frac{1}{n} = \frac{1}{2} + \frac{1}{3} + \ldots = \infty$

Sum of the twins: $\sum_{\text{twins}} \frac{1}{n} = \left(\frac{1}{3} + \frac{1}{5}\right) + \left(\frac{1}{5} + \frac{1}{7}\right) + \left(\frac{1}{11} + \frac{1}{13}\right) + \ldots =$

Prime Numbers (Primzahlen)

Primes are infinite! Can we say (at least!) something about their asymptotic behaviour?

Prime Number Theorem $p_n \sim n \log n$

Asymptotic distribution law: $\pi(x) \sim \frac{x}{\log x}$

Sum of the inverses: $\sum_{primes\ n} \frac{1}{n} = \frac{1}{2} + \frac{1}{3} + ... = \infty$

Sum of the twins: $\sum_{\text{twins}} \frac{1}{n} = \left(\frac{1}{3} + \frac{1}{5}\right) + \left(\frac{1}{5} + \frac{1}{7}\right) + \left(\frac{1}{11} + \frac{1}{13}\right) + ... = 1.90216...$

Prime Numbers (Primzahlen)

Primes are infinite! Can we say (at least!) something about their asymptotic behaviour?

Prime Number Theorem $p_n \sim n \log n$

Asymptotic distribution law: $\pi(x) \sim \frac{x}{\log x}$

Sum of the inverses: $\sum_{primes\ n} \frac{1}{n} = \frac{1}{2} + \frac{1}{3} + ... = \infty$

Sum of the twins: $\sum_{\text{twins}} \frac{1}{n} = \left(\frac{1}{3} + \frac{1}{5}\right) + \left(\frac{1}{5} + \frac{1}{7}\right) + \left(\frac{1}{11} + \frac{1}{13}\right) + ... = 1.90216...$

Prime Numbers (Primzahlen)

Primes are infinite! Can we say (at least!) something about their asymptotic behaviour?

Prime Number Theorem $p_n \sim n \log n$

Asymptotic distribution law: $\pi(x) \sim \frac{x}{\log x}$

Sum of the inverses: $\sum_{primes\ n} \frac{1}{n} = \frac{1}{2} + \frac{1}{3} + \ldots = \infty$

Sum of the twins: $\sum_{\text{twins}} \frac{1}{n} = \left(\frac{1}{3} + \frac{1}{5}\right) + \left(\frac{1}{5} + \frac{1}{7}\right) + \left(\frac{1}{11} + \frac{1}{13}\right) + \ldots = 1.90216\ldots$

Riemann's z-function: $\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$

Prime Numbers (Primzahlen)

Primes are infinite! Can we say (at least!) something about their asymptotic behaviour?

Prime Number Theorem $p_n \sim n \log n$

Asymptotic distribution law: $\pi(x) \sim \frac{x}{\log x}$

Sum of the inverses: $\sum_{primes\ n} \frac{1}{n} = \frac{1}{2} + \frac{1}{3} + \ldots = \infty$

Sum of the twins: $\sum_{\text{twins}} \frac{1}{n} = \left(\frac{1}{3} + \frac{1}{5}\right) + \left(\frac{1}{5} + \frac{1}{7}\right) + \left(\frac{1}{11} + \frac{1}{13}\right) + \ldots = 1.90216\ldots$

Riemann's z-function: $\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s} = \prod_{p\ prime} \frac{1}{1 - p^{-s}}$

Complexity

When is a problem "difficult"? (Complexity Theory)

Question: How much TIME (or SPACE) does it take to solve a problem?

Assume:

- Every elementary operation takes the same amount of time
- We are interested in the asymptotic scaling as function of the input

Examples:

- Add N numbers: O(N)

When is a problem "difficult"? (Complexity Theory)

Question: How much TIME (or SPACE) does it take to solve a problem?

Assume:

- Every elementary operation takes the same amount of time
- We are interested in the asymptotic scaling as function of the input

Examples:

- Add N numbers: O(N)
- Calculate the determinant of an NxN matrix: O(N!)

When is a problem "difficult"? (Complexity Theory)

Question: How much TIME (or SPACE) does it take to solve a problem?

Assume:

- Every elementary operation takes the same amount of time
- We are interested in the asymptotic scaling as function of the input

Examples:

- Add N numbers: O(N)
- Calculate the determinant of an NxN matrix: O(N!)

  If you are smart: $O(N^3)$

When is a problem "difficult"? (Complexity Theory)

Question: How much TIME (or SPACE) does it take to solve a problem?

Assume:

- Every elementary operation takes the same amount of time
- We are interested in the asymptotic scaling as function of the input

Examples:

- Add N numbers: O(N)
- Calculate the determinant of an NxN matrix: O(N!)

If you are smart: $O(N^3)$

If you are VERY smart: $O(N^{2.373})$

When is a problem "difficult"? (Complexity Theory)

Question: How much TIME (or SPACE) does it take to solve a problem?

Assume:

- Every elementary operation takes the same amount of time
- We are interested in the asymptotic scaling as function of the input

Examples:

- Add N numbers: O(N)
- Calculate the determinant of an NxN matrix: O(N!)

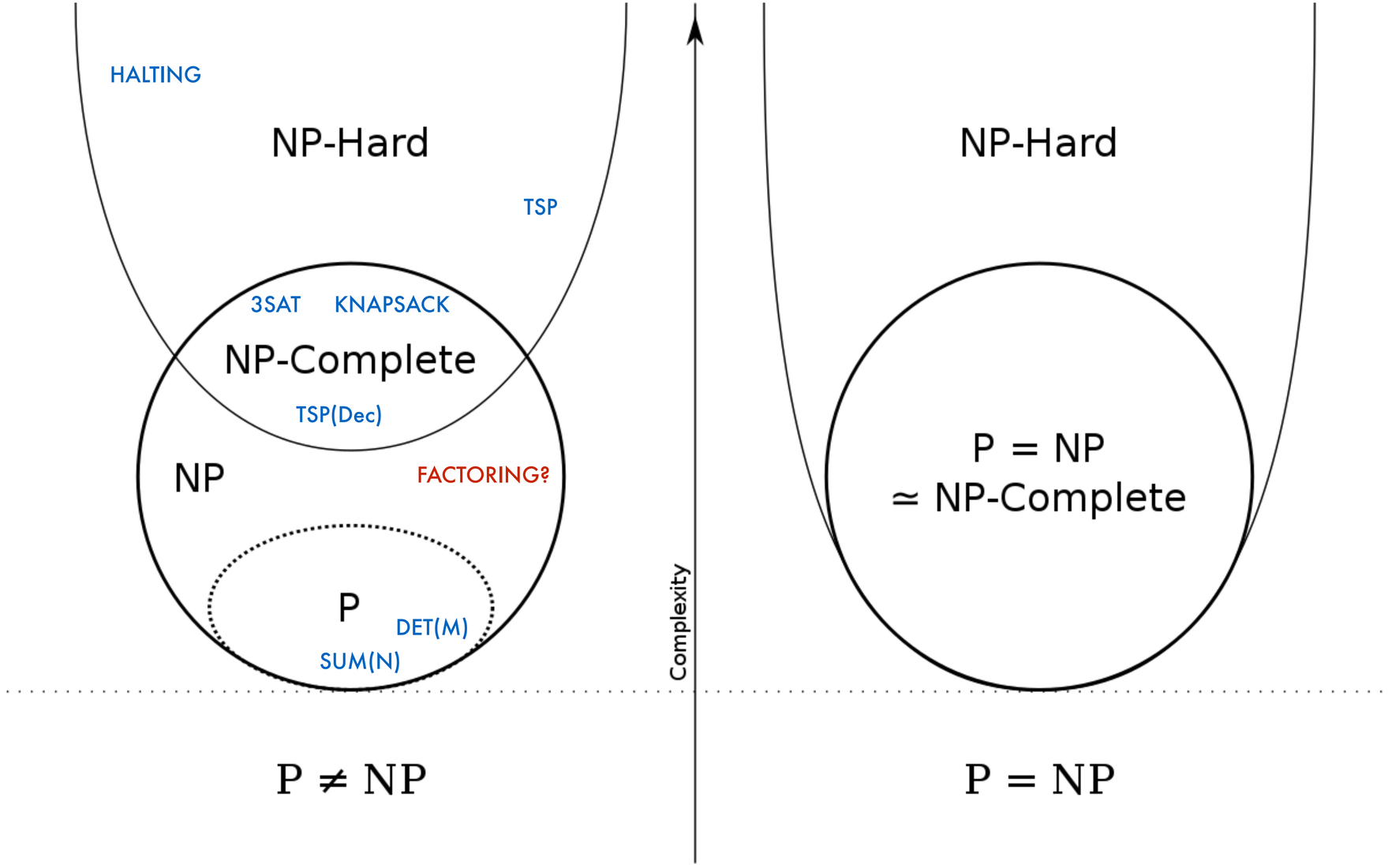
If you are smart: $O(N^3)$

If you are VERY smart: $O(N^{2.373})$

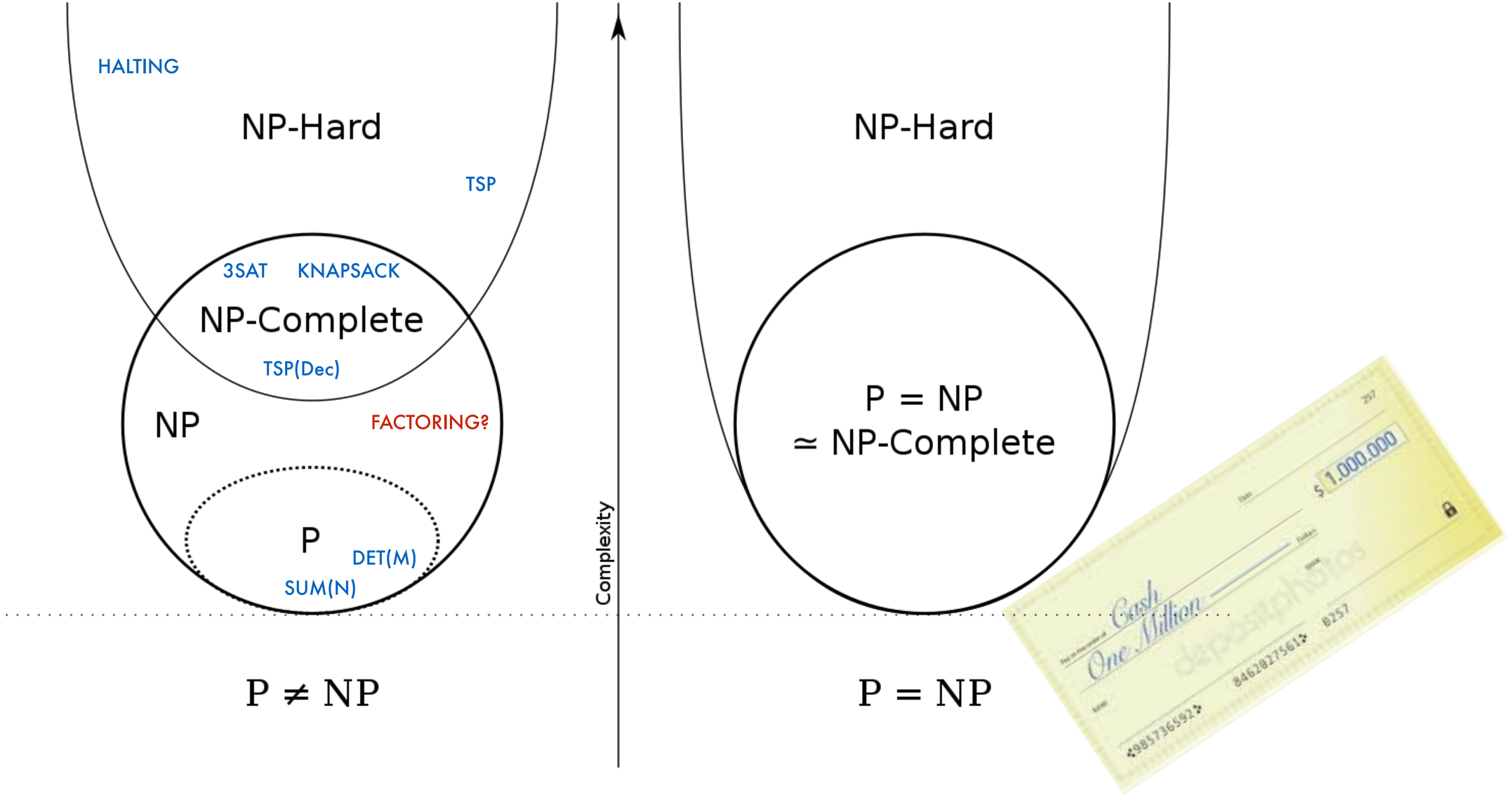
What about "non-polynomial" problems? Example:

Travelling salesman problem (TSP): visit N cities in a loop taking the shortest path

When is a problem "difficult"? (Complexity Theory)



When is a problem "difficult"? (Complexity Theory)



Factorization: a “difficult” problem?

Consider two prime numbers p and q and calculate N = qp

- If you know N, can you factorize it (discover q and p)?
- How “difficult” is this?

Example: N = 35

Factorization: a “difficult” problem?

Consider two prime numbers p and q and calculate N = qp

- If you know N, can you factorize it (discover q and p)?
- How “difficult” is this?

Example: N = 35 p = 5 , q = 7 (wow..)

Factorization: a "difficult" problem?

Consider two prime numbers p and q and calculate N = qp
- If you know N, can you factorize it (discover q and p)?
- How "difficult" is this?

Example: N = 35 p = 5 , q = 7 (wow..)

What about this?
1.444.363

Should you try all the combinations of (prime) divisors?

Factorization: a "difficult" problem?

Consider two prime numbers p and q and calculate N = qp
- If you know N, can you factorize it (discover q and p)?
- How "difficult" is this?

Example: N = 35 p = 5 , q = 7 (wow..)

What about this?
1.444.363

Should you try all the combinations of (prime) divisors?

It is believed that FACTORING is in NP (but not NP-complete!)

Factorization: a "difficult" problem?

Consider two prime numbers p and q and calculate N = qp
- If you know N, can you factorize it (discover q and p)?
- How "difficult" is this?

Example: N = 35 p = 5 , q = 7 (wow..)

What about this?
1.444.363

Should you try all the combinations of (prime) divisors?

BTW: p = 1181 , q = 1223

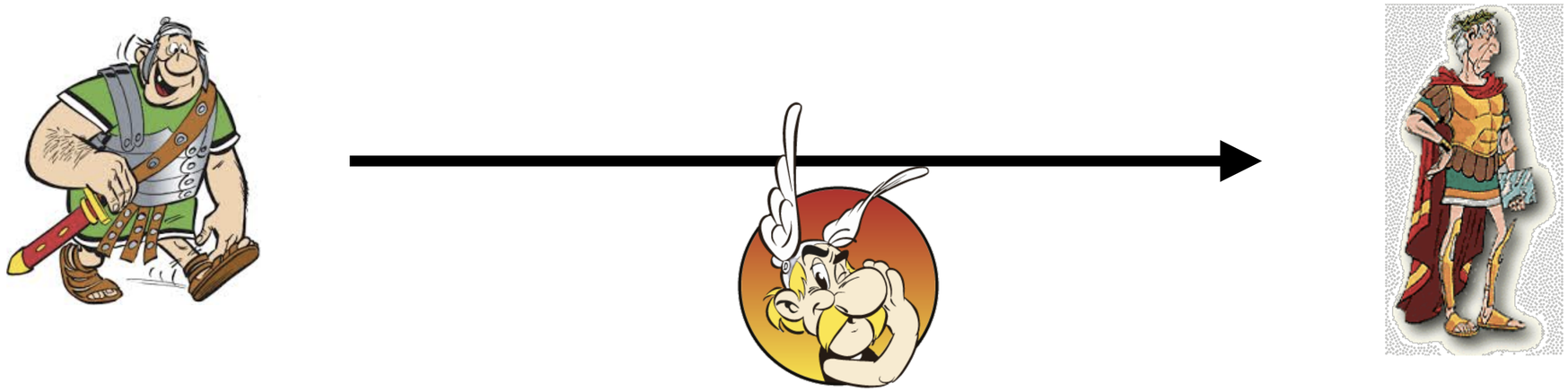
It is believed that FACTORING is in NP (but not NP-complete!)

Cryptography

Back to Roman Times

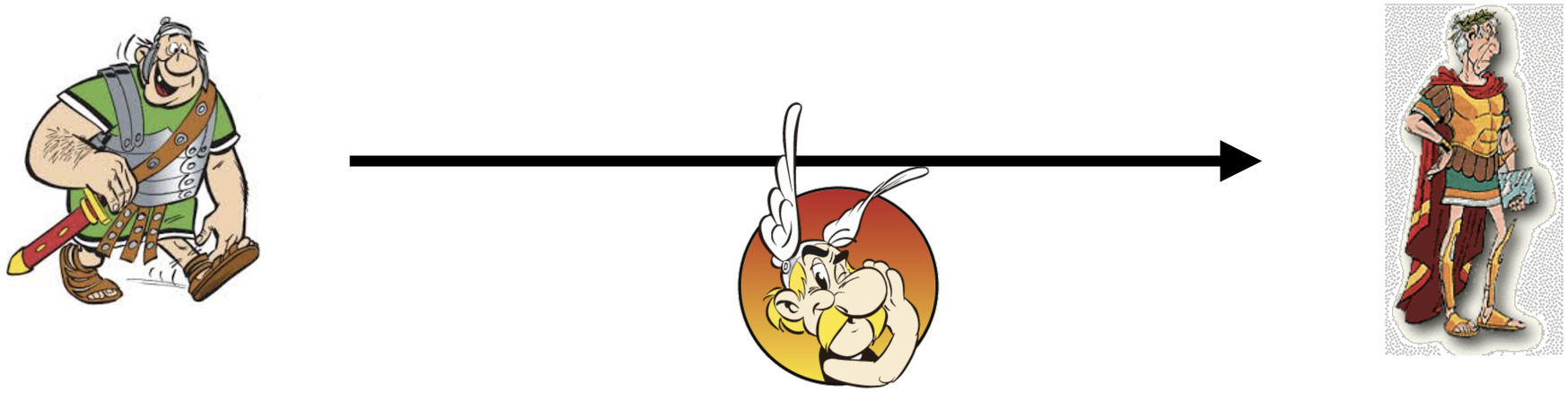
The “Caesar cipher” is one of the oldest forms of cryptography.
Idea: Shift a letter by 3 steps in the alphabet: A—>D, B—>F, …, Z—>C

Back to Roman Times



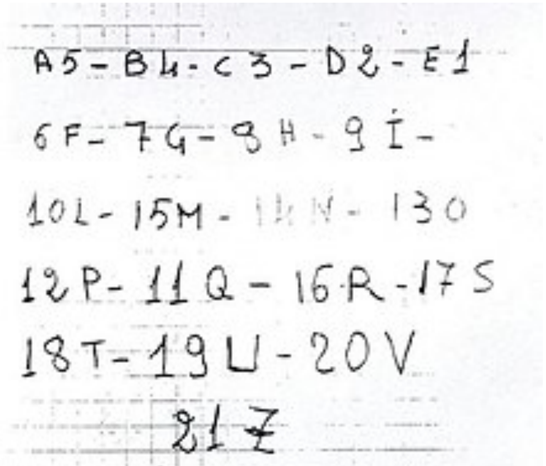
The “Caesar cipher” is one of the oldest forms of cryptography.
Idea: Shift a letter by 3 steps in the alphabet: A—>D, B—>F, …, Z—>C

Back to Roman Times



The “Caesar cipher” is one of the oldest forms of cryptography.
Idea: Shift a letter by 3 steps in the alphabet: A—>D, B—>F, …, Z—>C

Apparently still in use until 2006…



https://de.wikipedia.org/wiki/Pizzino

Cryptography with Primes: RSA (simplified)

Rivest, R.; Shamir, A.; Adleman, L. Comm. ACM. 21 (2): 120–126 (1978).

You

Bank

Choose p,q primes

N=pq

Find d (private key):

$3d=1 \bmod ((p-1)(q-1))$

d from Extended Euclidean Algorithm

Most commonly: 3—> 65537

Cryptography with Primes: RSA (simplified)

Rivest, R.; Shamir, A.; Adleman, L. Comm. ACM. 21 (2): 120–126 (1978).

You

Credit card number X

Calculate $c = X^3 \bmod(N)$

"easy" (modular exponentiation)

(N,3)
(public key)

Bank

Choose p,q primes

N=pq

Find d (private key):

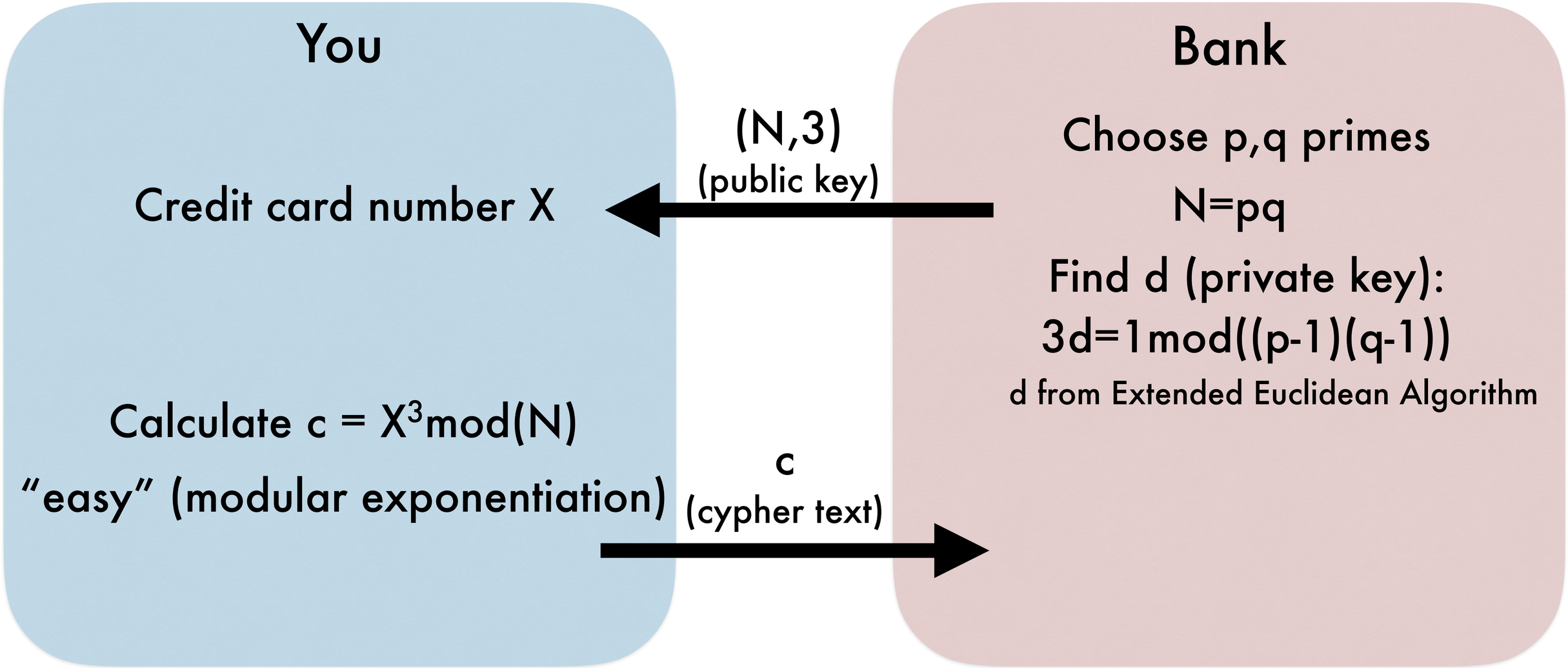
$3d=1 \bmod((p-1)(q-1))$

d from Extended Euclidean Algorithm

Most commonly: 3—> 65537

Cryptography with Primes: RSA (simplified)

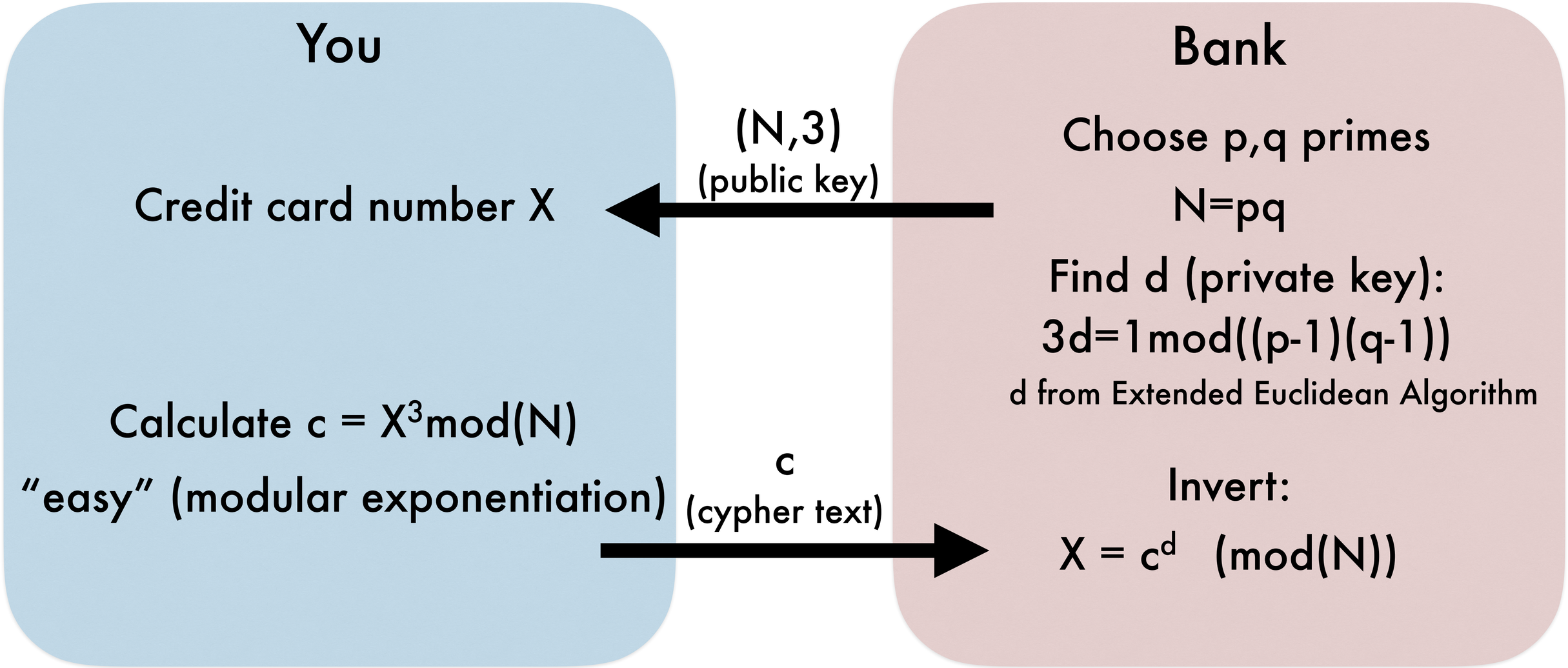
Rivest, R.; Shamir, A.; Adleman, L. Comm. ACM. 21 (2): 120–126 (1978).



Most commonly: 3—> 65537

Cryptography with Primes: RSA (simplified)

Rivest, R.; Shamir, A.; Adleman, L. Comm. ACM. 21 (2): 120–126 (1978).



Most commonly: 3—> 65537

Cryptography with Primes: RSA (simplified)

Rivest, R.; Shamir, A.; Adleman, L. Comm. ACM. 21 (2): 120–126 (1978).

You

Credit card number X

Calculate $c = X^3 \bmod(N)$

"easy" (modular exponentiation)

Bank

Choose p,q primes

N=pq

Find d (private key):

$3d=1 \bmod((p-1)(q-1))$

d from Extended Euclidean Algorithm

Invert:

$X = c^d \pmod{N}$

(N,3)
(public key)

c
(cypher text)

Most commonly: 3–> 65537

Cryptography with Primes: RSA (simplified)

Rivest, R.; Shamir, A.; Adleman, L. Comm. ACM. 21 (2): 120–126 (1978).

You

Credit card number X

Calculate $c = X^3 \bmod(N)$

"easy" (modular exponentiation)

Bank

Choose p,q primes

N=pq

Find d (private key):

$3d = 1 \bmod((p-1)(q-1))$

d from Extended Euclidean Algorithm

Invert:

$X = c^d \pmod{N}$

(N,3)
(public key)

c
(cypher text)

Modular exponentiation is hard to invert.

Most commonly: 3–> 65537

Cryptography with Primes: RSA (simplified)

Euler's Theorem: $gdc(a, N) = 1 \Rightarrow a^{\phi(N)} = 1(modN) \qquad N = pq$

$$\phi(N) = (p-1)(q-1)$$

Period of the modular exponential function

$$a^{\phi(N)} modN = 1 \Rightarrow a^{\phi(N)} = kN + 1 \Rightarrow a^{\phi(N)+1} = kNa + a \Rightarrow \boxed{a^{\phi(N)+1} = a(modN)}$$

$\phi(N)$ is the PERIOD of the function $f(i) = a^i modN$

Factorization

$$(a^{\phi(N)} - 1)modN = 0 \Rightarrow (a^{\phi/2} + 1)(a^{\phi/2} - 1)modN = 0$$

N has common factors with $(a^{\phi/2} \pm 1) \quad \Rightarrow gdc(N, a^{r/2} \pm 1)$

Cryptography with Primes: RSA (simplified)

Euler's Theorem: $gdc(a, N) = 1 \Rightarrow a^{\phi(N)} = 1(modN) \qquad N = pq$

$$\phi(N) = (p-1)(q-1)$$

Period of the modular exponential function

$$a^{\phi(N)} modN = 1 \Rightarrow a^{\phi(N)} = kN + 1 \Rightarrow a^{\phi(N)+1} = kNa + a \Rightarrow \boxed{a^{\phi(N)+1} = a(modN)}$$

$\phi(N)$ is the PERIOD of the function $f(i) = a^i modN$

Factorization

$$(a^{\phi(N)} - 1)modN = 0 \Rightarrow (a^{\phi/2} + 1)(a^{\phi/2} - 1)modN = 0$$

N has common factors with $(a^{\phi/2} \pm 1) \quad \Rightarrow gdc(N, a^{r/2} \pm 1)$

Factorize N $\Longleftrightarrow$ Find the period of f

Cryptography with Primes: RSA (simplified)

Encrypt $c = x^3 mod N$ Public Key $(N, 3)$

Decrypt $x = c^d mod N$ Private Key $3d mod (p-1)(q-1) = 1$

Knowing the private key d, the inversion really works and is fast:

$$x = (x^3 mod N)^d mod N = x^{3d} mod N = x^{k\phi+1} = x^{k\phi} x mod N = x$$

Euler's Theorem

An eavesdropper can do the same factorizing N in p and q.
OR: he can try to find the period of the modular exp. function: classically hard.

Quantum Mechanics

Quantum mechanics (vs Probability Theory)

Classical probability theory:

Event with N possible outcomes: $(p_1, p_2, \ldots, p_N)$

1-norm: $\sum_i p_i = 1 \quad p_i \geq 0$ conserved by stochastic matrices (columns add to 1)

Quantum mechanics (vs Probability Theory)

Classical probability theory:

Event with N possible outcomes: $(p_1, p_2, \ldots, p_N)$

1-norm: $\sum_i p_i = 1 \quad p_i \geq 0$ conserved by stochastic matrices (columns add to 1)

Quantum Mechanics:

$$|\psi\rangle = \alpha_1|1\rangle + \alpha_2|2\rangle + \ldots + \alpha_N|N\rangle$$

2-norm: $\sum_i |\alpha_i|^2 = 1$ conserved by unitary matrices ("operators")

constructive/destructive interference

WF "collapse": only one component results from a measurement

Quantum Computers

Quantum Computers

1st Conference on Physics and Computation (MIT, 1981)

1918-1988

Simulating Physics with Computers

Richard P. Feynman

Department of Physics, California Institute of Technology, Pasadena, California 91107

Received May 7, 1981

Quantum Computers

1st Conference on Physics and Computation (MIT, 1981)

1918-1988

Simulating Physics with Computers

Richard P. Feynman

Department of Physics, California Institute of Technology, Pasadena, California 91107

Received May 7, 1981

Quantum Computers

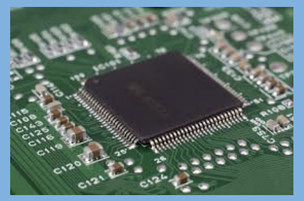
Classic register

0 1

Classic gates

Any functionally complete set of logic gates

(AND,NOT) (NAND) (NOR)...



Quantum Computers

Classic register

0 1

Quantum register

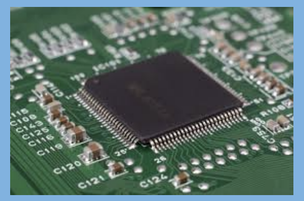
$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad |\alpha|^2 + |\beta|^2 = 1$$

Classic gates

Any functionally complete set of logic gates

(AND,NOT) (NAND) (NOR)...



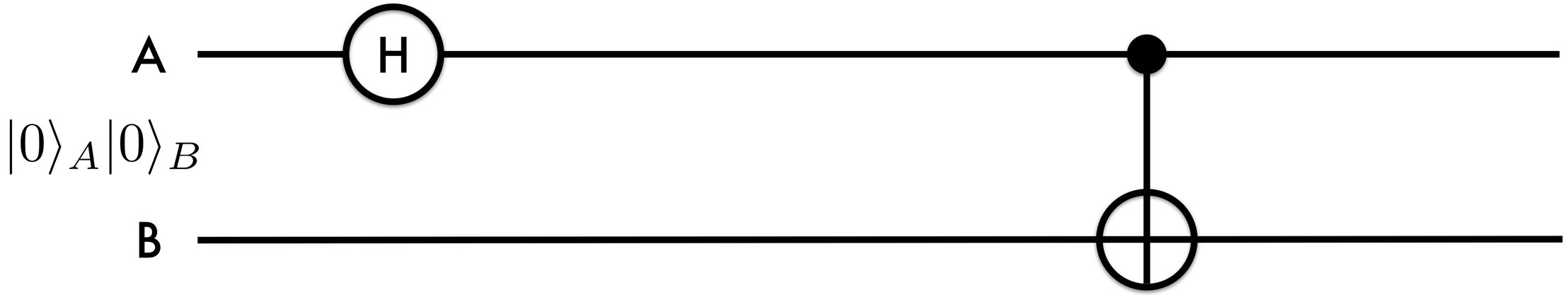
Quantum gates

Any functionally complete set of unitary operators

Example: Hadamard gate:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

Quantum Circuits



$$\frac{1}{\sqrt{2}}(|0\rangle_A| + |1\rangle_A)|0\rangle_B \qquad \frac{1}{\sqrt{2}}(|0\rangle_A|0\rangle_B + |1\rangle_A|1\rangle_B)$$

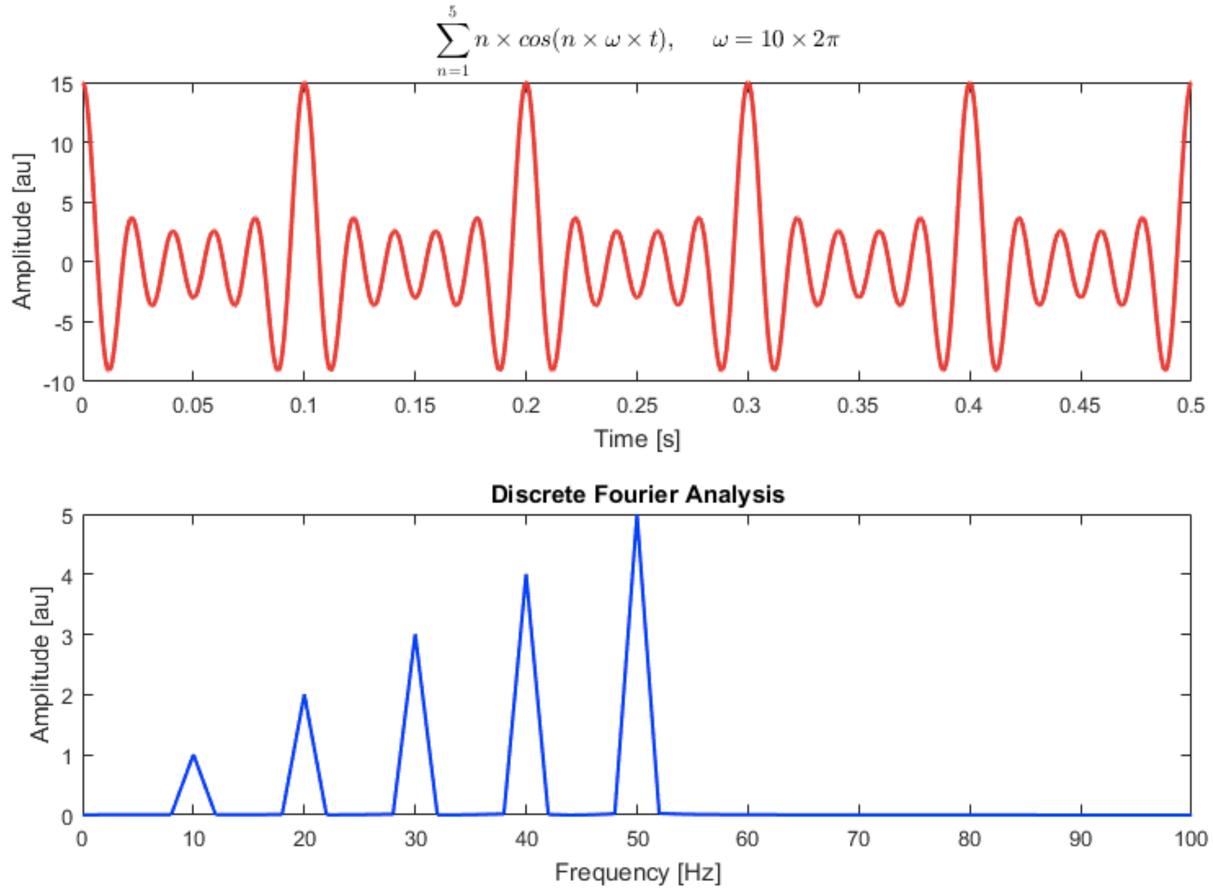
$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Swaps the qbit if the other one is |1>

Period Finding: Fourier Transform

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N} kn}$$
$$= \sum_{n=0}^{N-1} x_n \cdot \left[\cos\left(\frac{2\pi}{N} kn\right) - i \cdot \sin\left(\frac{2\pi}{N} kn\right)\right]$$



Quantum Fourier Transform

$$\sum_j \alpha_j |j\rangle \rightarrow \sum_k \tilde{\alpha}_k |k\rangle$$

$$\tilde{\alpha}_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i j k/N} \alpha_j$$

Example: $|10\rangle \rightarrow |00\rangle - |01\rangle + |10\rangle - |11\rangle$

$$F_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix}$$

$$\omega_N = e^{\frac{2\pi i}{N}}$$

Quantum Fourier Transform

$$\sum_j \alpha_j |j\rangle \rightarrow \sum_k \tilde{\alpha}_k |k\rangle \qquad \tilde{\alpha}_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i j k/N} \alpha_j$$

Example: $|10\rangle \rightarrow |00\rangle - |01\rangle + |10\rangle - |11\rangle$

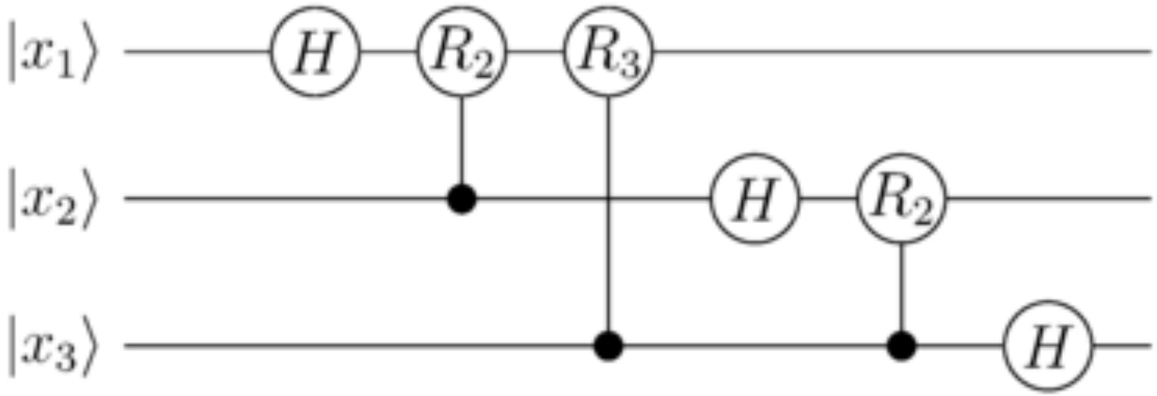
$$F_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix} \qquad \omega_N = e^{\frac{2\pi i}{N}}$$

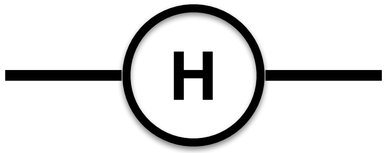
Can be implemented iteratively on a QC with two gates!

FFT: $O(n2^n)$ gates QFT: $O(n^2)$ gates

Quantum Fourier Transform

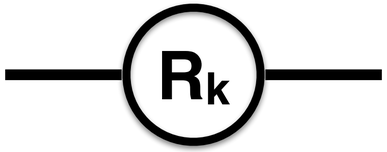
3-qubits example:





$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

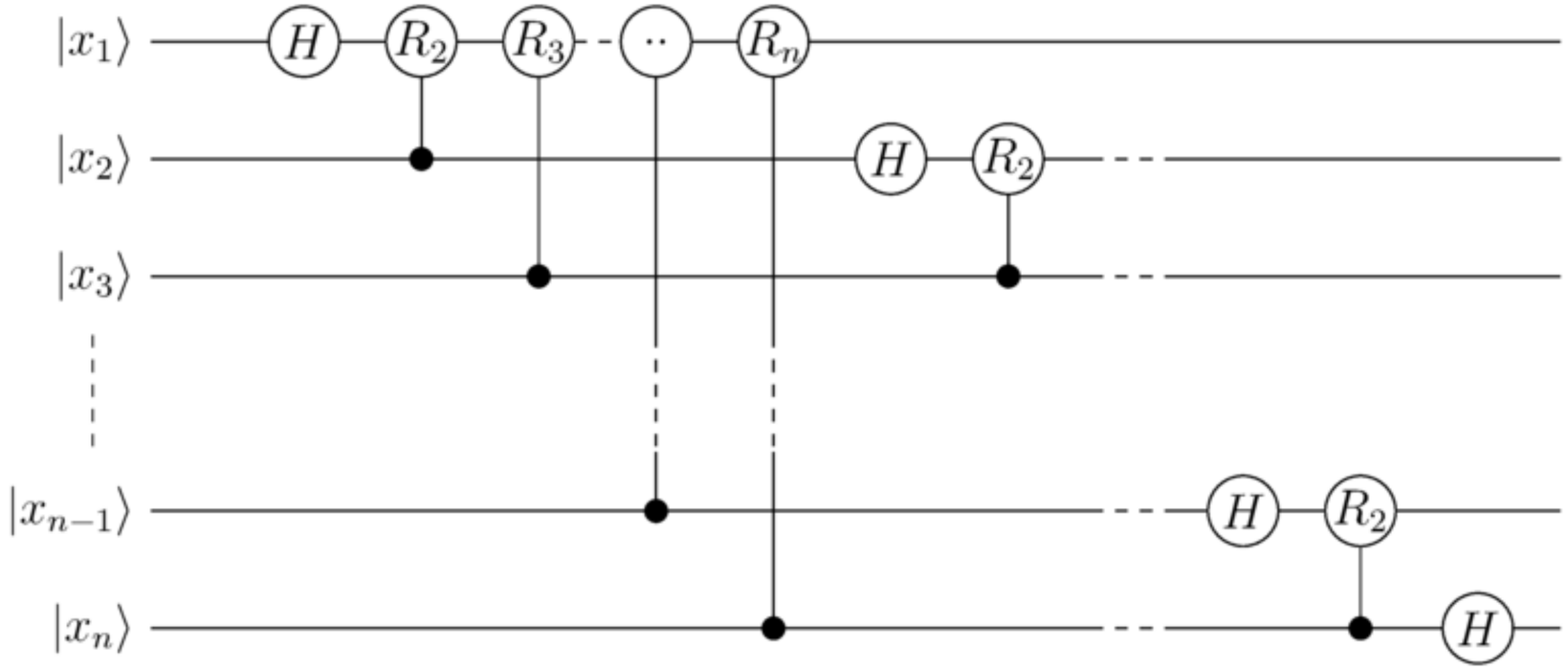
Hadamard Gate



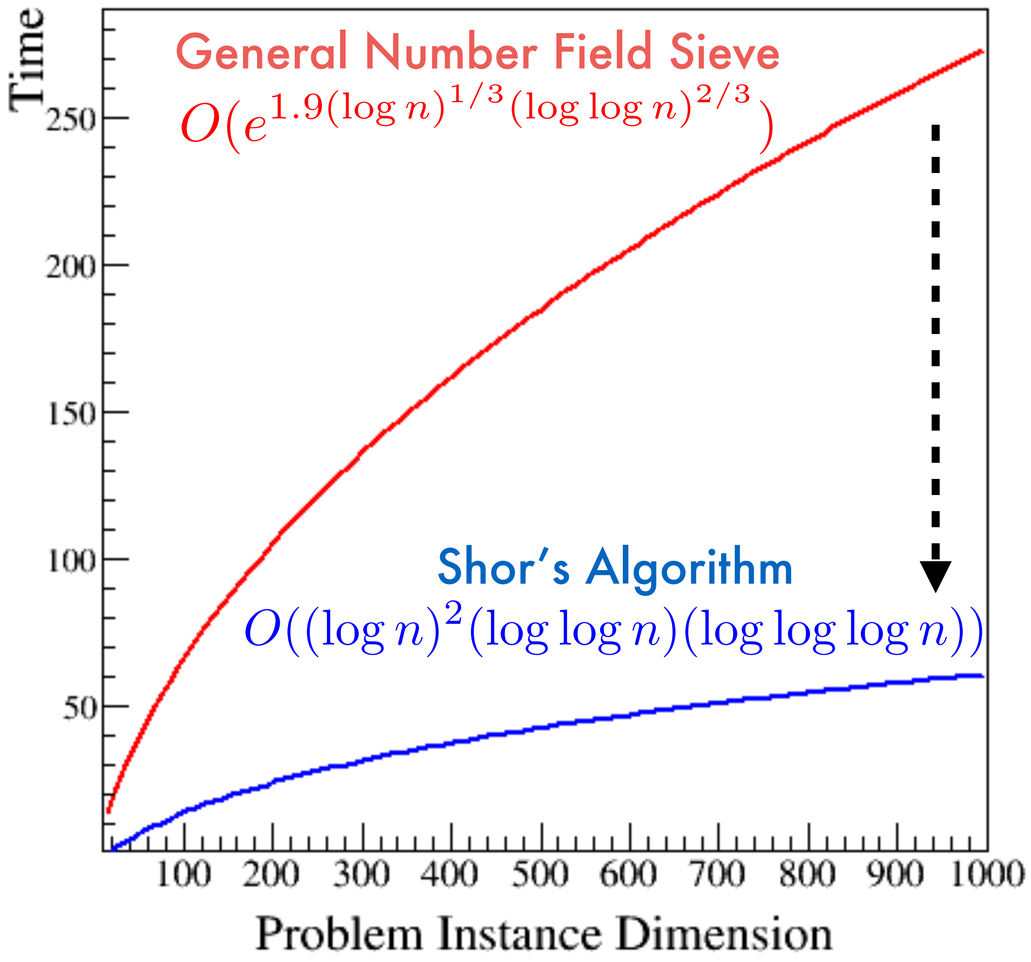
$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{pmatrix}$$

Phase-shift Gate

Quantum Fourier Transform



Shor's Algorithm

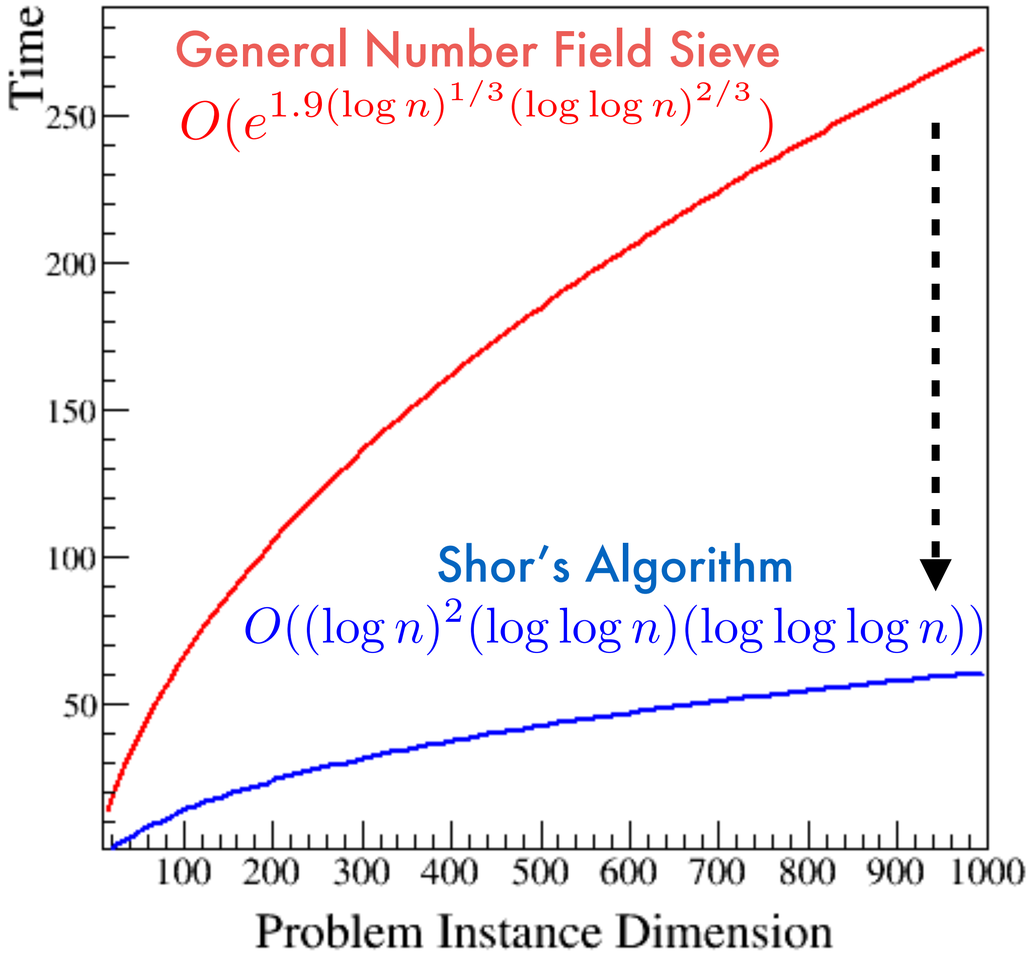


Classical computer:
240-digit number factored (2019)
900 cores/years

Peter Shor (1959,-)

P.W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring", Proc. 35th ann. symposium on Foundations of Computer Science. IEEE Comput. Soc. Press: 124-134.

Shor's Algorithm



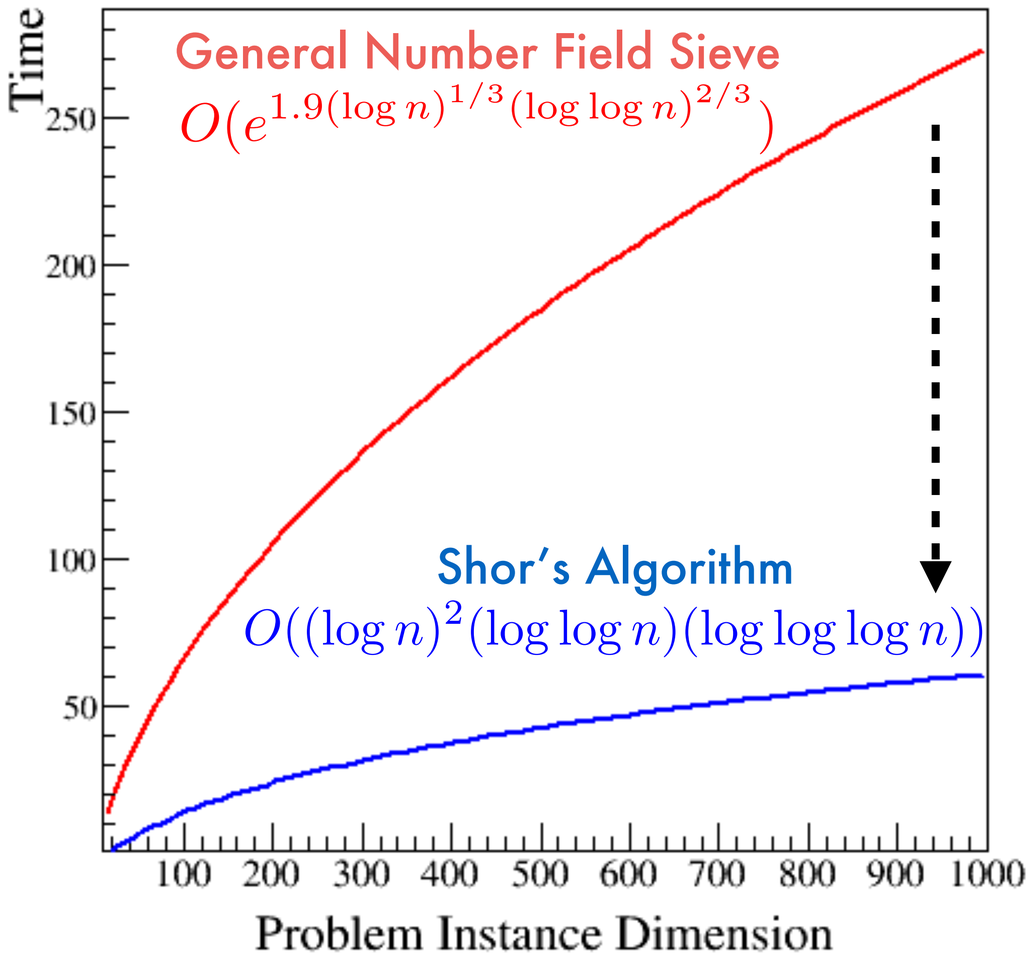
Classical computer:
240-digit number factored (2019)
900 cores/years

Peter Shor (1959,-)

15=5x3 (2001)

P.W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring", Proc. 35th ann. symposium on Foundations of Computer Science. IEEE Comput. Soc. Press: 124-134.

Shor's Algorithm



Classical computer:
240-digit number factored (2019)
900 cores/years

Peter Shor (1959,-)

15=5x3 (2001)

21=3x7 (2012)

P.W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring", Proc. 35th ann. symposium on Foundations of Computer Science. IEEE Comput. Soc. Press: 124-134.

Quantum Supremacy

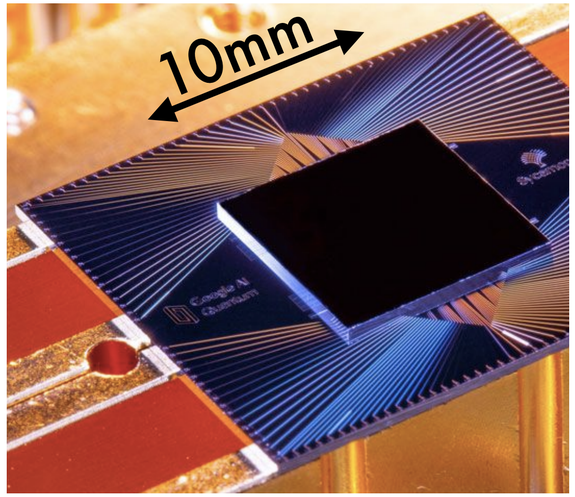
QS: Show that a quantum computer can be asymptotically faster than a classical one on a specific task.

Done in 2019 for the first time:

Google, with the Sycamore chip based on 54 q-bits ("transmons", a sort of non-linear quantum resonators in the few-GHz range realized with Josephson junctions) arranged on a 2D lattice.

Classically "Hard" problem:

Given a randomly chosen quantum circuit, predict the output distribution over the possible bit strings. This is very hard to simulate for a classical computer.

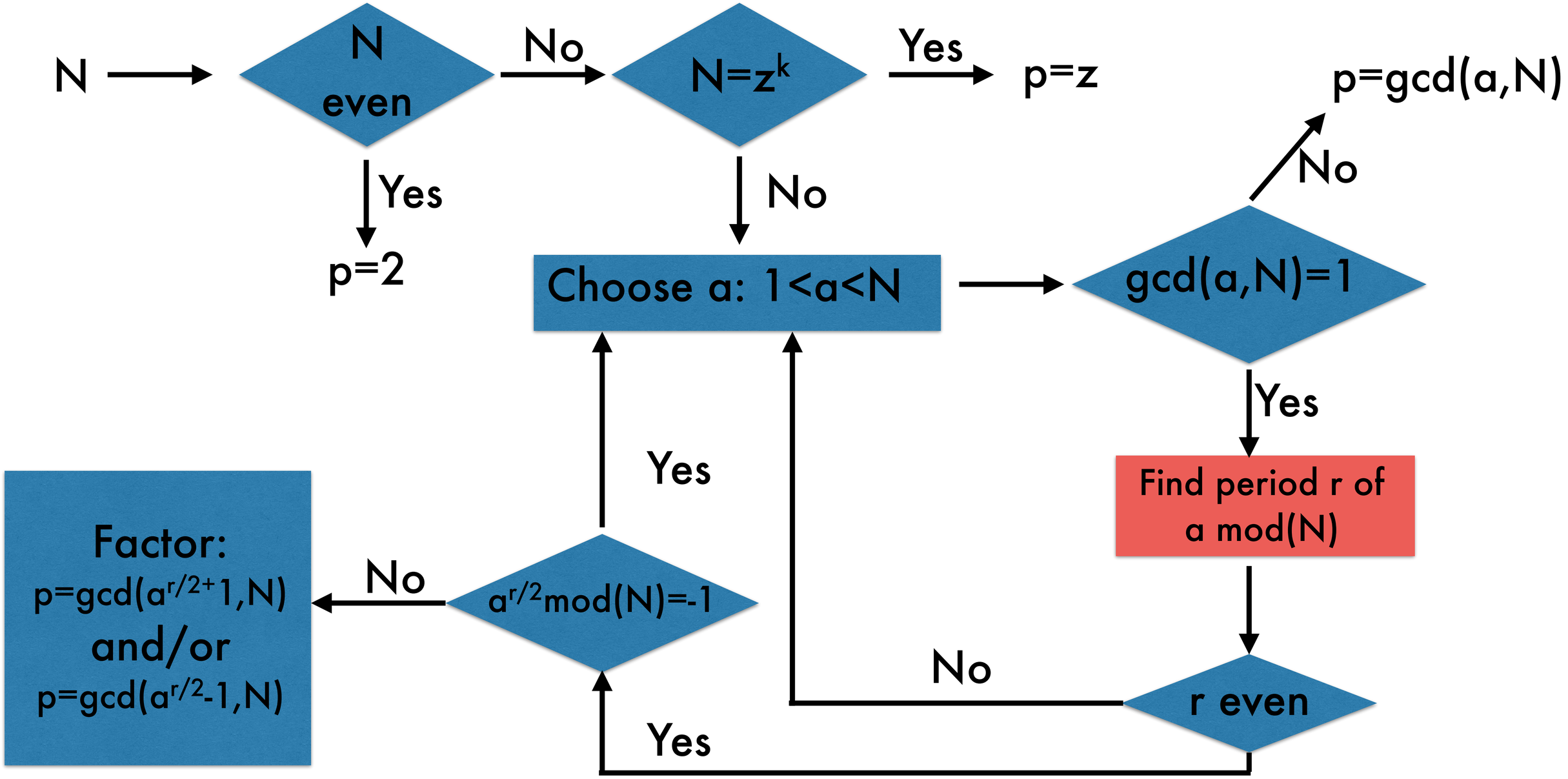


Summary

- QC: Asymptotically faster on certain (NP) problems.
- Cannot solve NP-complete problems!
- Many attempts at construction a QC: different technologies.
- No, a QC does not try all the possible solutions in parallel: quantum interference.
- Strong interest from industry (Google, IBM, Microsoft, D-Wave, SW companies..)

Vielen Dank fuer Ihre Aufmerksamkeit!
Thank you for your attention!

Factoring: General Algorithm



When is a problem "difficult"? (Complexity Theory)

A VERY informal intro to Complexity Classes

P: Decision problems solvable by a TM in polynomial time.

NP: Decision problems "hard" to solve but verifiable in polynomial time.

NP-hard: Any NP problem can be efficiently reduced to a problem in this class.

NP-complete: An NP-hard problem which is also in NP: "hardest problems in NP".

PSPACE: Problems which need polynomial space (memory!)

Much more: EXPTIME, EXPSPACE, PSPACE-complete, EXPTIME-complete, …