A-III

(3) - Period finding pb and Shor's factoring algorithm

* factoring pb :

Given a positive composite integer N, what prime numbers, when multiplied together are equal to it ?

here : $N = p\,q$

Best randomized classical algo :

$$O\left(\exp\left(c \cdot n^{1/3} (\log n)^{2/3}\right)\right)$$

where n = length of the bit string representing N.

Shor's quantum algorithm (1994) :

$$O(n^2 \log n \log\log n) = \Theta(n^2)$$

=> in principle, a QC can factor numbers exponentially faster than classical computers.

→ Why is Shor's factoring algo so imptt ?

* first example of a pb $\in$ BPQ, while believed not to be in BPP _

* anticipated implications for cryptography.
(=> could impact computer security, privacy ----)

Aside : Public key cryptography and the RSA (Rivest - Shamir - Adleman) system.

Cryptography = art of enabling two parties to communicate privately.
(e.g. credit card payment)

To achieve privacy a "cryptosystem" is used.

An effective cryptosystem should make it easy for the 2 parties to communicate but should make it hard for a third party to "eavesdrop" _

ex : public key cryptosystems

the idea is to avoid the exchange of secret key which could be intercepted and copied.

instead there is a public key accessible by anyone wants to send a msg to the receiver. (This public key is used to encrypt the msg)

and a private key needed to decrypt the msg, which is possessed by receiver only.

Most widely used public key cryptosyst is the RSA system, based on the (apparent) difficulty of factoring on a classical computer.

Ex: Suppose Alice & Bob want to communicate privately. In particular Alice wants to send a message to Bob.
Here is the procedure:

1) Bob selects two large prime numbers p & q and computes the product $N = pq$. (usually 1048 bits)

2) Bob computes the Euler function
$$\varphi(N) = N - p - q + 1$$
$$= (p-1)(q-1)$$

which is easy to compute if p, q are known, but hard otherwise.

($\varphi(N)$ = number of numbers $\leq N$ that are coprime with N)

3) Bob chooses randomly $e < \varphi(N)$ such that e is coprime with $\varphi(N)$

↳ $GCD(e, \varphi(N)) = 1$

4) Bob computes

$$d = e^{-1} \pmod{\varphi(N)}$$

(which can be done efficiently (Euclidean algo) if $\varphi(N)$ is known – hard otherwise)

5) Bob announces the public key $= \{N, e\}$ and retains the private key $= d$

6) Now Alice wants to send a message M (number $M < N$)

She encrypts it using the public key

$$M \longmapsto b = f(M) = M^e \pmod{N}$$

(can be done efficiently with repeated squaring)

7) Bob decrypts it by :

$$b \longmapsto b^d \pmod N$$

$$= M^{ed} \pmod N$$

$ed = 1 \pmod{\varphi(N)}$ $\Big\}$ $= M \cdot \left(M^{\varphi(N)}\right)^{\text{integer}} \pmod N$

$M^{\varphi(N)} = 1 \pmod N$ $\Big\}$ $= M \pmod N$

(Euler thm)

Now if another party ("Eve") can factor N

i.e. if Eve can determine p, q from N

$\Rightarrow$ Eve can then efficiently determine $\varphi(N)$ and $d = e^{-1} \pmod{\varphi(N)}$

$\Rightarrow$ thus if Eve intercepts the message b, she can break RSA and decrypt b.

$\Rightarrow$ the RSA system is based on the fact that factoring is hard to do on a classical computer.

and the Shor's (quantum) factoring algo could threaten the foundation of computer security -

Probably still a long time before this happens.
At the moment Shor's algo was used to factor the number 15 (in 2001) and 21 (in 2012). *and the number 35 recently. Recent status can be found in: Willsch et al, arXiv:2410.14397 (2024)*

But at some point will have to develop quantum cryptographic syst which would be safe against quantum attacks.

* Formulation of factoring as a period-finding (or "order-finding") pb :

We want to find the factors p, q of the composite number $N > 1$.
(note: N should be odd, not prime, not prime power)

(1) We randomly choose an integer $a \in \{2, \dots, N-1\}$ which is coprime to N.

↙ means that $GCD(a, N) = 1$

" greatest common divisor (largest positive integer that divides both a and N)

(if $GCD(a,N) = c \neq 1 \Rightarrow N = c \times$ integer
and we have solved the pb)

(2) Consider the fct $f_{N,a}(x) = a^x \pmod N$

with $x = 0, 1, 2 \ldots$

This fct is a sequence :

$a^0 \pmod N$, $a^1 \pmod N$, $a^2 \pmod N$ ....
$\|$
1

and this sequence will cycle after a while:

(this is because the numbers $a < N$ coprime with N form a finite group under multiplication mod $N \to \mathbb{Z}_N^*$)

$\exists \quad 0 < r \leq N$ such that

$a^r \equiv 1 \pmod N$

r is the period of the fct $f_{N,a}(x)$
(aka the order of the element a
of the multiplicative group $\mathbb{Z}_N^*$)

(non zero)
r is the smallest integer that satisfies
$a^r \equiv 1 \pmod N$.

$\to$ find r.

(3) if r is even :

define $X \equiv a^{r/2} \pmod N$

N divides $(X+1)(X-1)$

$a^r = 1 \pmod N$

$\Leftrightarrow \underbrace{(a^{r/2})^2}_{X^2} = 1 \pmod N$

$\Leftrightarrow (X+1)(X-1) = 0 \pmod N$

$\Leftrightarrow (X+1)(X-1) =$ integer $\times N$.

We know that N does not divide $(X-1)$

if it did we would have $X - 1 = 0 \pmod N$

$X = 1 \pmod N$

$X = a^{r/2}$

i.e. the period (order of a) would be $r/2$ (at most).

and we suppose that N also does not divide $(X+1)$. $(X \neq -1 \pmod N)$

In that case N must have a non trivial factor common with $(X+1)$ and a non trivial factor common with $(X-1)$.

$\Rightarrow$ p and/or q will be contained in

$\{ GCD(X+1, N) \; ; \; GCD(X-1, N) \}$

$\Rightarrow$ The overall protocol for factoring N into two prime numbers $N = pq$ is:

(1) pick $a < N$, coprime with N

(2) find the order r of a (aka the period of $f_{a,N}(x) = a^x \pmod N$)

(3) if r is even

$$X \equiv a^{r/2} \pmod N$$

if $X + 1 \not\equiv 0 \pmod N$

$\Rightarrow$ compute $p, q = GCD(X \pm 1, N)$

else: try another a.

Note: finding the GCD of 2 integers x, y $(x > y)$ is easy, using $GCD(x,y) = GCD(x-y, y)$ or $GCD(x,y) = GCD(x \bmod y, y)$ (Euclidean algo)

Example : factorization of $N = 15$

(1) pick $a = 13$

(2) find the period of the sequence

13^0 13^1 13^2 13^3 $\boxed{13^4}$ 13^5 ....

$=$ $=$ $=$ $=$ $=$

1 13 169 2197 28561

$=$ $=$ $=$

$4 \pmod{15}$ $7 \pmod{15}$ $1 \pmod{15}$

$\Rightarrow$ $r = 4$

(3) $X = a^{r/2} = 13^2 = 169 \pmod{15}$

$= 4 \pmod{15}$

$GCD(X+1, N) = GCD(5, 15) = 5$

$GCD(X-1, N) = GCD(3, 15) = 3.$

$\Rightarrow$ $p = 5, \quad q = 3.$

$\Rightarrow$ Thus the pb of factoring reduces to finding the period r of the function given by the modular exponentiation

$$f_{a,N}(x) = a^x \pmod N$$

Note :

in general the period-finding pb is the following -

We are given some fct f

$$f : \{0,1\}^n \longrightarrow \{0,1\}^m$$

such that $f(y) = f(x)$ iff $y = x + j \cdot r$ (integer)

ie $y = x \pmod r$

and the goal is to find the period r.

Classically this is believed to be a hard pb.

Even if one can efficiently compute $f(x)$, the period could be exponential in n and thus it would take an exponential number of queries to find r.

However it can be done efficiently with a quantum circuit using QFT.

In the case where f is the modular exponentiation (factoring pb)

$$f_{a,N}(x) = a^x \pmod N$$

we will now see that the quantum circuit is just the circuit for QPE with the unitary operator U chosen such that

$$U_a|x\rangle = |ax \pmod N\rangle$$

where $|x\rangle$ = L-qubit comput-basis state, $L = \log(N)$

Note: the period-finding pb for fct $f_{a,N}(x) = a^x \pmod N$ is called "order-finding pb."

if the order of a is r $\Rightarrow$ $a^r = 1 \pmod N$

$$\Rightarrow (U_a)^r |x\rangle = |a^r x \pmod N\rangle$$

$$= |x\rangle$$

$$\Rightarrow (U_a)^r = \mathbb{1}$$

Thus all eigenvalues of U_a are r-th roots of unity:

$$\lambda_k = e^{2i\pi k/r} \qquad k = \{0, 1, \ldots r-1\}$$

The corresponding eigenstates are:

$$|\psi_k\rangle = \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} e^{-(2i\pi)kj/r} \; |a^j \pmod N\rangle$$

check:

$$U_a |\psi_k\rangle = \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} e^{-(2\pi i)kj/r} \underbrace{U_a \, |a^j \pmod N\rangle}_{=\; |a^{j+1} \pmod N\rangle}$$

$j' = j+1$

$$= \frac{1}{\sqrt{r}} \sum_{j'=1}^{r} e^{-(2\pi i)k(j'-1)/r} \; |a^{j'} \pmod N\rangle$$

$$= \frac{1}{\sqrt{r}} \sum_{j'=0}^{r-1} e^{-(2\pi i)kj'/r} \, e^{2\pi i k/r} \; |a^{j'} \pmod N\rangle$$

$$= e^{2\pi i k/r} \, |\psi_k\rangle$$

used the fact that the term $(j' = r)$ is:

$$e^{-(2\pi i)k(r-1)/r} \, |a^r \pmod N\rangle$$

$$\underbrace{e^{-(2\pi i)k}}_{1} \; e^{+(2\pi i)k/r} \; |a^0 \pmod N\rangle$$

because $a^r = 1 \pmod N$
$= a^0 \pmod N$

$= \text{term}\ (j' = 0)$

Thus, the Quantum Phase Estimation procedure allows us to obtain $\theta \approx \frac{k}{r}$, to high accuracy.

Shor's factoring algorithm (continued)

There are two requirements to be able to apply QPE:

1) we must be able to implement efficiently the controlled-U^{2^j} operations, for any integer j.

A priori since 2^j is exponential in j we might think that this cannot be done in polynomial time.

But in fact calculating $a^{2^j} \pmod N$ efficiently is possible via the (classical) "repeated squaring method":

$$a^{2^j} \pmod N = \left(a^{2^{j-1}}\right)^2 \pmod N$$

since $j = 0, \dots t-1$ $\Rightarrow$ we only need t multiplications to compute all the a^{2^j} and each require $\sim (\log N)^2 = L^2$ elementary operations.

we will choose $t = (2L+1) + \log\left(2 + \frac{1}{2\epsilon}\right)$ (see further why) $\Rightarrow O(L^3)$ operations in total.

2) we must be able to prepare an eigenstate $|\psi_k\rangle$ of U_a, or at least a superposition of such eigenstates.

$|\psi_k\rangle$ would require to know r $\rightarrow$ impossible.
But we have:

$$\frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |\psi_k\rangle$$

$$= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} e^{-(2i\pi)kj/r} \; |a^j \pmod N\rangle$$

$$= \sum_{j=0}^{r-1} \underbrace{\left(\frac{1}{r} \sum_{k=0}^{r-1} e^{-2i\pi kj/r} \right)}_{S_{j,0}} |a^j \pmod N\rangle$$

$$= |a^0 \pmod N\rangle = |1\rangle$$

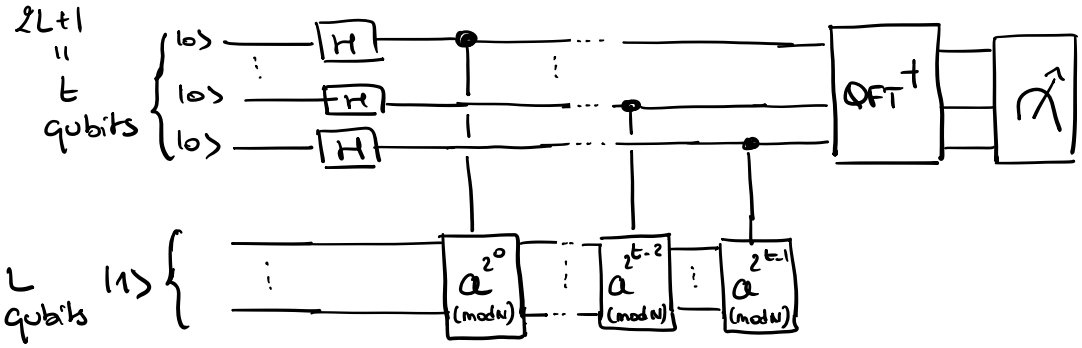
which is trivial to construct.

If we use $t = 2L+1 + \log\left(2 + \frac{1}{2\epsilon}\right)$ qubits in the top register, and prepare the bottom register in state $|1\rangle$

$\Rightarrow$ we will obtain an estimate θ_k of k/r accurate to $2L+1$ bits with proba at least $\frac{1-\epsilon}{r}$ -

(according to previous lecture).

$\Rightarrow$ Circuit for order finding :



$O(L)$ gates $\quad O(L^3)$ gates $\quad O(L^2)$

$\Rightarrow O(L^3)$

The QPE circuit above provide $\theta_k \approx k/r$ from which we can obtain the order r with a little bit more work.

Explanations of how it is done can be found in the Nielsen and Chuang -

In brief : There is a theorem stating that if k/r is rational and $\left|\frac{k}{r} - \theta_k\right| \leq \frac{1}{2r^2}$ then one can find r using the continued fraction algorithm in $O(L^3)$ operat°.

(appendix 4_ Nielsen & Chuang)

here we have $\left|\frac{k}{r} - \theta_k\right| \leq 2^{-2L-1} \leq \frac{1}{2r^2}$

because $r \leq N \leq 2^L$ - And $\frac{k}{r}$ is a priori rational (ratio of 2 bounded integers) -

$\Rightarrow$ the theorem applies.

The idea of the continued fraction algo is to describe real numbers in terms of integers using expressions of the form

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\cdots + \cfrac{1}{a_n}}}}}$$

where the a_i's are positive integers.

The continued fraction expression of a rational number terminates at some pt.

Note: it might happen that u and r have a common factor, in which case the continued fraction algo could return r' a multiple of r, instead of r itself.

But there are ways around this issue (see Nielsen Chuang p 229 - 231) ensuring proba of success of order 1.