

Quantum Algorithms

QUANTUM FOURIER TRANSFORM

The Discrete Fourier Transform (DFT) takes a complex vector x_0, \dots, x_{N-1} (of length N) into another complex vector y_0, \dots, y_{N-1} such that

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi i jk/N}.$$

The Quantum Fourier Transform takes an orthonormal basis $|0\rangle, \dots, |N-1\rangle$ into a new one given by states

$$|j\rangle \longrightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i jk/N} |k\rangle.$$

QFT is linear and unitary operator. The action on an arbitrary state is

$$\sum_{j=0}^{N-1} x_j |j\rangle \longrightarrow \sum_{k=0}^{N-1} y_k |k\rangle,$$

so that y_k are DFT of x_j .

We take $N = 2^n$ and the basis $|0\rangle \dots, |2^n - 1\rangle$ for a n qubit system. Furthermore,

$$\begin{aligned} j &= j_1 2^{n-1} + j_2 2^{n-2} + \dots + j_n 2^0 \implies \\ j &\longrightarrow j_1 j_2 \dots j_n \end{aligned}$$

and we introduce the *binary fraction*

$$0.j_1 j_{l+1} \dots j_m := \frac{j_1}{2} + \frac{j_{l+1}}{4} + \dots + \frac{j_m}{2^{m-l+1}}.$$

Then

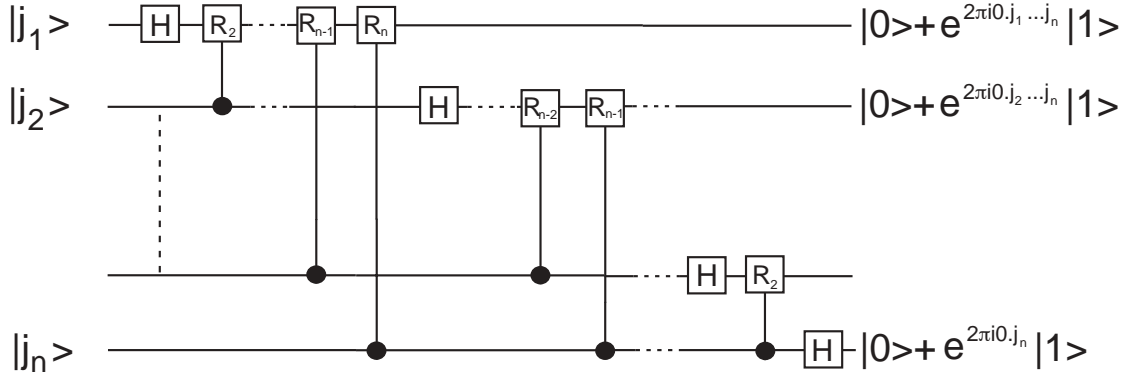
$$\begin{aligned}
|j\rangle &\longrightarrow \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} \exp\left(2\pi i j \frac{k}{2^n}\right) |k\rangle \\
&= \frac{1}{2^{n/2}} \sum_{k_1=0}^1 \cdots \sum_{k_n=0}^1 \exp\left(2\pi i j \left(\sum_{l=1}^n k_l 2^{-l}\right)\right) |k_1 \dots k_n\rangle \\
&= \frac{1}{2^{n/2}} \sum_{k_1=0}^1 \cdots \sum_{k_n=0}^1 \bigotimes_{l=1}^n \exp\left(2\pi i j \frac{k_l}{2^l}\right) |k_l\rangle \\
&= \frac{1}{2^{n/2}} \bigotimes_{l=1}^n \left[\sum_{k_l=0}^1 \exp\left(2\pi i j \frac{k_l}{2^l}\right) |k_l\rangle \right] \\
&= \frac{1}{2^{n/2}} \bigotimes_{l=1}^n \left[|0\rangle + e^{2\pi i j / 2^l} |1\rangle \right] \\
&= \frac{1}{2^{n/2}} \left(|0\rangle + e^{2\pi i 0 \cdot j_n} |1\rangle \right) \left(|0\rangle + e^{2\pi i 0 \cdot j_{n-1} j_n} |1\rangle \right) \cdots \left(|0\rangle + e^{2\pi i 0 \cdot j_1 j_2 \cdots j_n} |1\rangle \right) .
\end{aligned}$$

With the notation

$$U_{\phi_k} \longrightarrow R_k \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i / 2^k} \end{bmatrix}$$

the QFT can be implemented with the following circuit

FIGURES



Here swap gates are omitted.

The sequence of operations from left to right are:

$$\begin{aligned}
 |j_1 \dots j_n\rangle &\longrightarrow H \text{ to the first qubit} \\
 \frac{1}{2^{1/2}} \left(|0\rangle + e^{2\pi i \cdot j_2} |1\rangle \right) |j_1 \dots j_n\rangle &\longrightarrow CR_2 \\
 \frac{1}{2^{1/2}} \left(|0\rangle + e^{2\pi i \cdot j_1 j_2} |1\rangle \right) |j_2 \dots j_n\rangle &\longrightarrow CR_3 \dots CR_n \\
 \frac{1}{2^{1/2}} \left(|0\rangle + e^{2\pi i \cdot j_1 j_2 \dots j_n} |1\rangle \right) |j_2 \dots j_n\rangle &
 \end{aligned}$$

The same procedure on the 2nd qubit leads to

$$\frac{1}{2^{2/2}} \left(|0\rangle + e^{2\pi i \cdot j_1 j_2 \dots j_n} |1\rangle \right) \left(|0\rangle + e^{2\pi i \cdot j_2 \dots j_n} |1\rangle \right) |j_3 \dots j_n\rangle .$$

Continuing in this fashion for each qubit we obtain

$$\frac{1}{2^{n/2}} \left(|0\rangle + e^{2\pi i \cdot j_1 j_2 \dots j_n} |1\rangle \right) \left(|0\rangle + e^{2\pi i \cdot j_2 \dots j_n} |1\rangle \right) \dots \left(|0\rangle + e^{2\pi i \cdot j_n} |1\rangle \right) .$$

Then, swap operations reverse the order of qubits

$$\frac{1}{2^{n/2}} \left(|0\rangle + e^{2\pi i \cdot j_n} |1\rangle \right) \left(|0\rangle + e^{2\pi i \cdot j_{n-1} j_n} |1\rangle \right) \dots \left(|0\rangle + e^{2\pi i \cdot j_1 j_2 \dots j_n} |1\rangle \right) .$$

How many gates does the circuit use?

$$H + (n - 1) CR \quad \text{on the first qubit} - \text{total of } n \text{ gates}$$

$$H + (n - 2) CR \quad \text{on the second qubit} - \text{total of } (n - 1) \text{ gates}$$

⋮

$$n(n - 1)/2 \text{ gates}$$

Then, at most $n/2$ swaps are required (each swap can be accomplished by three *CNOT*).

Thus the circuit provides a $\Theta(n^2)$ algorithm for QFT \rightarrow efficient!

In contrast, classical algorithms for computing DFT on 2^n elements use $\Theta(n2^n)$ gates (like Fast Fourier Transform)!

But, drawback the amplitudes in a quantum computer cannot be directly accessed by measurement, thus, the use of QFT is not so trivial.

DEUTSCH'S ALGORITHM

Suppose we are given an oracle that upon request computes a function

$$f : \{0, 1\} \mapsto \{0, 1\}.$$

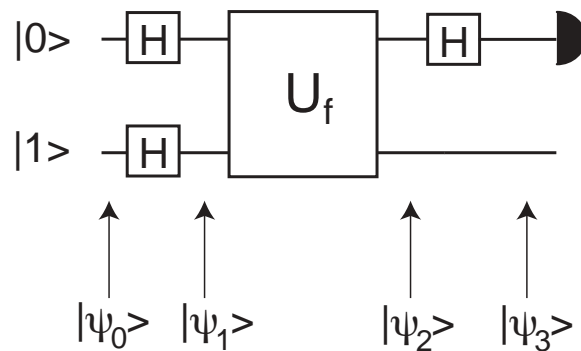
Assume that f is either constant or balanced

$$\left. \begin{array}{l} f(0) = f(1) = 0 \\ \text{or } f(0) = f(1) = 1 \end{array} \right\} \text{ constant}$$

$$\left. \begin{array}{l} f(0) = 0, f(1) = 1 \\ \text{or } f(0) = 1, f(1) = 0 \end{array} \right\} \text{ balanced.}$$

Classically, to decide whether f is constant or balanced we need to evaluate $f(0)$ and $f(1)$ (two queries).

Quantum only one query!



$$|\psi_0\rangle = |01\rangle,$$

$$|\psi_0\rangle = \frac{1}{2} (|0\rangle + |1\rangle) (|0\rangle - |1\rangle).$$

Notice that

$$|x\rangle (|0\rangle - |1\rangle) \xrightarrow{U_f} |x\rangle |f(x)\rangle - |x\rangle |1 \oplus f(x)\rangle = |x\rangle (-1)^{f(x)} (|0\rangle - |1\rangle),$$

then

$$|\psi_2\rangle = \begin{cases} \pm \frac{1}{2} (|0\rangle + |1\rangle) (|0\rangle - |1\rangle) & \text{if } f(0) = f(1) \\ \pm \frac{1}{2} (|0\rangle - |1\rangle) (|0\rangle - |1\rangle) & \text{if } f(0) \neq f(1) \end{cases}$$

and

$$|\psi_3\rangle = \begin{cases} \pm \frac{1}{\sqrt{2}} |0\rangle (|0\rangle - |1\rangle) & \text{if } f(0) = f(1) \\ \pm \frac{1}{\sqrt{2}} |1\rangle (|0\rangle - |1\rangle) & \text{if } f(0) \neq f(1) \end{cases}$$

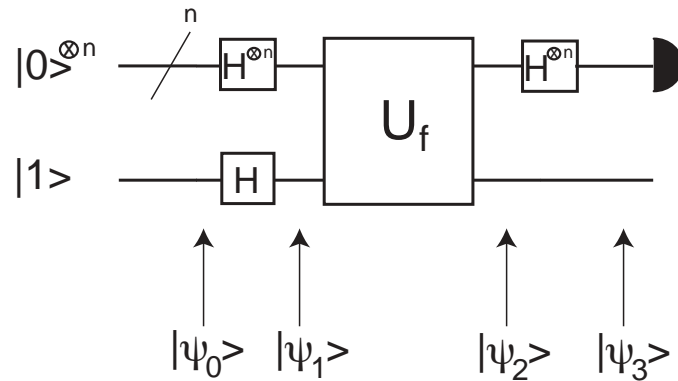
Finally, by measuring the first qubit in the computational basis it is possible to establish whether f is constant or balanced.

DEUTSCH-JOZSA ALGORITHM

$$f : \{0, 1\}^n \mapsto \{0, 1\}$$

$$f \begin{cases} \text{constant} & f(x) = 0 \text{ (or } f(x) = 1) \forall x \in \{0, 1\}^n \\ \text{balanced} & f(x) = 1 \text{ for half possible values of } x \text{ and } f(x) = 0 \text{ for the other half} \end{cases}$$

Classically one needs of $2^{n-1} + 1$ queries to establish the type of f .



$$\begin{aligned}
|\psi_0\rangle &= |0\rangle^{\otimes n} |1\rangle \\
|\psi_1\rangle &= \sum_{x \in \{0,1\}^n} \frac{|x\rangle}{\sqrt{2^n}} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \\
|\psi_1\rangle &= \sum_x \frac{(-1)^{f(x)} |x\rangle}{\sqrt{2^n}} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right]
\end{aligned}$$

Notice that

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_z (-1)^{x \cdot z} |z\rangle,$$

then

$$|\psi_3\rangle = \sum_z \sum_x \frac{1}{2^n} [(-1)^{x \cdot z + f(x)} |z\rangle] \frac{1}{\sqrt{2}} [|0\rangle - |1\rangle].$$

Now consider $|z\rangle = |0\rangle^{\otimes n}$

- if f is constant, then the amplitude for $|0\rangle^{\otimes n}$ is ± 1 depending on the value $f(x)$ takes; all other amplitudes are zero;
- if f is balanced, then the positive and negative contributions to the amplitude for $|0\rangle^{\otimes n}$ cancel.

Summarizing, if the measurement yields all 0s then the function is constant, otherwise the function is balanced.

QUANTUM SEARCH ALGORITHM (GROVER)

Suppose we wish to search through a search space of N (unstructured) elements. Rather than search the elements directly, we concentrate on the index $0, \dots, N - 1$. We assume $N = 2^n$ and that the problem has exactly M solutions.

The problem can be represented by a function f which takes as input $x \in \{0, N - 1\}$, and $f(x) = 1$ if x is a solution while $f(x) = 0$ if x is not a solution.

Suppose we are supplied with an oracle O with the ability to recognize solutions to the problem.

Classically we need of $O(N/M)$ oracle calls!

Quantum?

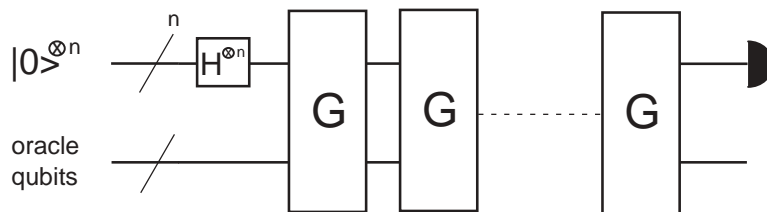
Recall the Deutsch-Jozsa algorithm

$$|x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \xrightarrow{O} (-1)^{f(x)} |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right).$$

Since the oracle qubit is not changed we can write

$$|x\rangle \xrightarrow{O} (-1)^{f(x)} |x\rangle.$$

The quantum search algorithm consists of repeated application of a quantum subroutine know as Grover iteration (G)

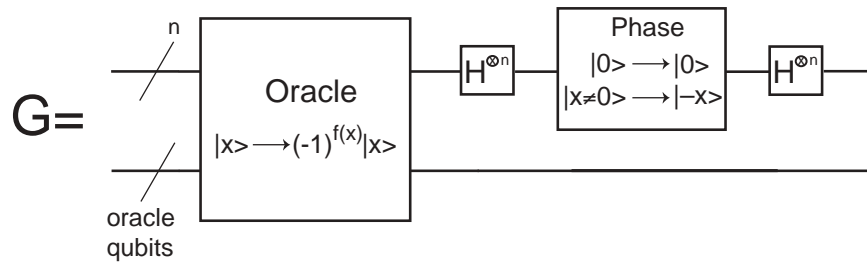


The algorithm begins with the state $|0\rangle^{\otimes n}$, then Hadamard transforms the state into

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle,$$

then G is applied repeatedly.

Grover operator



1. Apply the oracle O .
2. Apply $H^{\otimes n}$.
3. Perform conditional phase shift on the computer qubits, so that

$$|x\rangle \longrightarrow -(-1)^{\delta_{x0}} |x\rangle.$$

As conditional operation on n qubits this requires $O(n)$ gates. Furthermore, the unitary operator corresponding to this phase shift is given by $2|0\rangle\langle 0| - I$, in fact

$$(2|0\rangle\langle 0| - I) |x\rangle = 2\langle 0|x\rangle|0\rangle - |x\rangle = \begin{cases} |0\rangle & x = 0 \\ -|x\rangle & x \neq 0 \end{cases}$$

4. Apply $H^{\otimes n}$.

The combined effect of steps 2, 3, 4 is

$$H^{\otimes n} (2|0\rangle\langle 0| - I) H^{\otimes n} = 2|\psi\rangle\langle\psi| - I.$$

Thus

$$G = (2|\psi\rangle\langle\psi| - I) O.$$

Geometric interpretation

Denote

\sum_x' sum over all x solutions of the problem,

\sum_x'' sum over all x not solutions of the problem.

Define

$$|\alpha\rangle \equiv \frac{1}{\sqrt{N-M}} \sum_x'' |x\rangle, \quad |\beta\rangle \equiv \frac{1}{\sqrt{M}} \sum_x' |x\rangle,$$

then

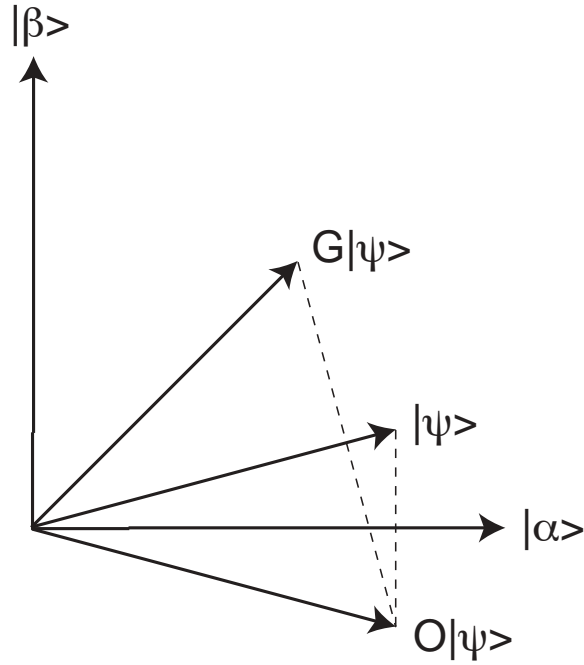
$$|\psi\rangle = \sqrt{\frac{N-M}{N}} |\alpha\rangle + \sqrt{\frac{M}{N}} |\beta\rangle,$$

i.e. the initial state of the computer is in the space spanned by $|\alpha\rangle$ and $|\beta\rangle$.

$$O(a|\alpha\rangle + b|\beta\rangle) = a|\alpha\rangle - b|\beta\rangle \quad \text{reflection about vector } |\alpha\rangle,$$

$$2|\psi\rangle\langle\psi| - I \quad \text{reflection about vector } |\psi\rangle.$$

Thus: $G \equiv$ product of two reflections \equiv rotation



$$\text{Let } \cos \frac{\theta}{2} = \sqrt{\frac{N-M}{N}},$$

$$|\psi\rangle = \cos \frac{\theta}{2} |\alpha\rangle + \sin \frac{\theta}{2} |\beta\rangle \quad \text{then,}$$

$$G|\psi\rangle = \cos\left(\frac{3}{2}\theta\right)|\alpha\rangle + \sin\left(\frac{3}{2}\theta\right)|\beta\rangle \quad (\text{so, rotation angle} = \theta),$$

$$G^k|\psi\rangle = \cos\left(\frac{2k+1}{2}\theta\right)|\alpha\rangle + \sin\left(\frac{2k+1}{2}\theta\right)|\beta\rangle,$$

$$G^k|\psi\rangle \quad \text{remains in the space spanned by } |\alpha\rangle, |\beta\rangle \text{ for all } k.$$

How many times must G be repeated in order to rotate $|\psi\rangle$ near $|\beta\rangle$?

Rotating

$$|\psi\rangle = \sqrt{\frac{N-M}{N}}|\alpha\rangle + \sqrt{\frac{M}{N}}|\beta\rangle,$$

through $\arccos\left(\sqrt{M/N}\right)$ radians takes the system to $|\beta\rangle$.

Then, repeating G

$$R = \text{Closest Integer} \left(\frac{\arccos(\sqrt{M/N})}{\theta} \right)$$

times, rotates $|\psi\rangle$ to within an angle $\theta/2 \leq \pi/4$ of $|\beta\rangle$. Measurement in the computational basis yields the solution with probability at least $1/2$.

However

$$M \ll N \implies \theta \sim \sin\theta \sim 2\sqrt{M/N}$$

$$\text{error } \frac{\theta}{2} \sim \sqrt{\frac{M}{N}} \implies \text{prob of error} \sim \frac{M}{N}.$$

Furthermore, from the above, $R < \lceil \pi/2\theta \rceil$ and assuming $M < N/2$

$$\frac{\theta}{2} \geq \sin\frac{\theta}{2} = \sqrt{\frac{M}{N}} \implies R \leq \left\lceil \frac{\pi}{4} \sqrt{\frac{N}{M}} \right\rceil.$$

Thus, $O\left(\sqrt{N/M}\right)$ oracle calls must be performed to get the exact solution with high prob (a quadratic improvement over $O\left(\sqrt{N/M}\right)$ classical).

FACTORIZING ALGORITHM (SHOR)

In order to factor a number N we shall use quantum computation to solve an equivalent problem.

Given N , choose randomly a fixed number $y < N$ coprime to N [$\gcd(y, N) = 1$], then, the major task in the algorithm will be to find the period r of the function:

$$F_N(a) = y^a \bmod N.$$

Consider the quadratic equation

$$x^2 = 1 \bmod N$$

$x = \pm 1 \bmod N$ are trivial solutions.

If N is an odd prime p , then these are the only solutions.

If N is composite, then there are also pairs of non trivial solutions $x = \pm a \bmod N$.

▷ **Example:**

$$x^2 = 1 \bmod 341$$

$$\text{trivial solutions } x = \pm 1 \bmod 341$$

$$\text{non trivial solutions } x = \pm 32 \bmod 341 \quad (\text{since } 341 = 11 \times 31).$$

In a general case let $N = n_1 \cdot n_2$ with $\gcd(n_1, n_2) = 1$ and consider

$$\begin{array}{ll} a) \left\{ \begin{array}{l} x_1 = 1 \bmod n_1 \\ x_1 = 1 \bmod n_2 \end{array} \right. & b) \left\{ \begin{array}{l} x_2 = -1 \bmod n_1 \\ x_2 = -1 \bmod n_2 \end{array} \right. \\ c) \left\{ \begin{array}{l} x_3 = 1 \bmod n_1 \\ x_3 = -1 \bmod n_2 \end{array} \right. & d) \left\{ \begin{array}{l} x_4 = -1 \bmod n_1 \\ x_4 = 1 \bmod n_2 \end{array} \right. \end{array}$$

In each case $x_i^2 = 1 \pmod{n_1}$ and $\pmod{n_2}$; so each x_i satisfies $x^2 = 1 \pmod{N}$.

By the Chinese remainder theorem each set has a unique solution \pmod{N} .

From $a)$ and $b)$ we get

$$x_1 = 1 \text{ and } x_2 = -1 \pmod{N} \quad (\text{trivial solutions}),$$

and from $c)$ and $d)$ we get

$$x_3 = a \text{ and } x_4 = -a \pmod{N} \quad (\text{non trivial solutions}).$$

Thus, it is

$$(a + 1)(a - 1) = 0 \pmod{N}$$

and $\gcd(N, (a \pm 1))$ are non trivial factors of N !

Notice that \gcd can be found efficiently using Euclid's algorithm.

▷ **Example:**

$$x^2 = 1 \pmod{341},$$

non trivial solutions $x = \pm 32 \pmod{341}$

$\gcd(31, 341) = 31$ and $\gcd(33, 341) = 11$.

Given N choose randomly $y < N$. If y and N are coprime, then let r be the order of $y \pmod{N}$, i.e. the least power of y such that

$$y^r = 1 \pmod{N}.$$

This is precisely the period of the function F_N . If r is also even, then setting

$$x = y^{r/2}$$

we have $x^2 = 1 \pmod{N}$, hence x is a candidate for non trivial solution of equation $x^2 = 1 \pmod{N}$.

The above process may fail if the chosen y value has an odd order r , or if r is even, but $y^{r/2}$ turns out to be a trivial solution of equation $x^2 = 1 \pmod N$. However:

★ **Theorem:**

Let N be odd with prime factorization

$$N = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k} .$$

Suppose y is chosen randomly satisfying $\gcd(y, N) = 1$. Let r be the order of $y \pmod N$, then

$$\text{Prob} \left(r \text{ is even and } y^{r/2} \neq \pm 1 \pmod N \right) \geq 1 - \frac{1}{2^{k-1}} .$$

We have excluded even values of N , since all factors of 2 are easy to recognize and remove. Also pure prime powers $N = p^\alpha$ are easily recognized by efficient classical algorithm.

Notice that it is possible to show that if $1 \leq y \leq N$ is selected randomly, then

$$\text{Prob} (\gcd(y, N) = 1) \geq \frac{1}{\log N} .$$

Thus, by assuming to be able to find the period r of y , we obtain a non trivial factor of N with $\text{prob} \geq \frac{1}{2 \log N}$ from the above process.

▷ **Example** factoring $N = 15$

Select $y < 15$ so that $\gcd(y, 15) = 1$

$$\{ 2, 4, 7, 8, 11, 13, 14 \} .$$

Let us pick up $y = 11$

$$y^r = 1 \pmod{15} \implies r = 2$$

$$x = y^{r/2} = 11 \implies \gcd(x \pm 1, N) \begin{cases} 5 \\ 3 \end{cases}$$

In this particular example any choice of y , except $y = 14$, leads to the correct result.

Shor (1994) describes a quantum algorithm which provides the order r of a randomly chosen y .

Given N , choose $y = 2^L$ between N^2 and $2N^2$. Choose a random $y < N$ and begin with an L qubit register in the state $|0\rangle$. Apply Hadamard getting

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle.$$

Compute $y^a \bmod N$, storing the result in a second register, giving

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle |y^a \bmod N\rangle.$$

This can be done efficiently e.g. by repeatedly squaring mod N to get y^{2^i} and multiplying selected ones corresponding to the binary expansion of a .

Next perform a measurement on the second register. Suppose the result is z

$$z = y^l \bmod N \quad \text{for some least } l$$

also

$$y^l = y^{jr+l} \bmod N \quad \forall j$$

Thus, the measurement will select values

$$a = l, l + r, l + 2r, \dots, l + Ar,$$

where A is the greatest integer less than $(q - l)/r$.

Notice that $l \leq r$ and $q \sim O(N^2) \implies A \sim q/r$.

Post measurement state of 1st register

$$|\phi_l\rangle = \frac{1}{\sqrt{A+1}} \sum_{j=0}^A |jr+l\rangle.$$

Consider the simple case in which r divides q exactly, so $A = q/r - 1$ and

$$|\phi_l\rangle = \sqrt{\frac{r}{q}} \sum_{j=0}^{\frac{q}{r}-1} |jr + l\rangle = \sum_{a=0}^{q-1} f(a)|a\rangle.$$

$$f(a) = \begin{cases} \sqrt{r/q} & (a - l) \text{ multiple of } r \\ 0 & \text{otherwise} \end{cases}$$

Performing now the QFT we get

$$|\phi_l\rangle \mapsto \sum_c \tilde{f}(c)|c\rangle$$

where $\tilde{f}(c)$ is the DFT of $f(a)$,

$$\begin{aligned} \tilde{f}(c) &= \frac{\sqrt{r}}{q} \sum_{j=0}^{q/r-1} \exp\left(\frac{2\pi i(jr + l)c}{q}\right) \\ &= \frac{\sqrt{r}}{q} \underbrace{\left[\sum_{j=0}^{q/r-1} \exp\left(2\pi i \frac{jrc}{q}\right) \right]}_{=0 \text{ or } q/r \text{ if } c \text{ is a multiple of } q/r} \exp\left(2\pi i \frac{lc}{q}\right) \\ \tilde{f}(c) &= \begin{cases} \exp(2\pi ilc/q) / \sqrt{r} & c \text{ is a multiple of } q/r \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

Writing $c = jq/r$ the final state becomes

$$\frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} \exp\left(\frac{2\pi ilj}{r}\right) \left| j \frac{q}{r} \right\rangle.$$

Thus a measurement (1st register) of the state's labeled by c will yield a multiple $\lambda q/r$ with $\lambda = 0, \dots, r - 1$ chosen equiprobably.

Note that the initial shift l disappears due to the translational invariance property of the *QFT*.

After the measurement we know a value c satisfying $c/q = \lambda/r$ (q is also known). If $\gcd(\lambda, r) = 1$, we can determine r by canceling c/q down to an irreducible fraction.

★ Since λ is chosen at random, it is possible to show that

$$\text{Prob}(\gcd(\lambda, r) = 1) \geq \frac{1}{\log r} \quad \text{for largish } r.$$

Thus, repeat the computation $O(\log r) < O(\log N)$ guarantees a success prob close to 1, hence an efficient determination of r .

★ Notice that the fastest classical algorithms for factorization run in time of order

$$\exp \left[(\log N)^{1/3} (\log \log N)^{2/3} \right],$$

e.g. they need a couple of billion years to factorize a 200-digit number!

Appendix: The General Case

Let us return to the general case where r does not exactly divide q

$$\tilde{f}(c) = \frac{\sqrt{r}}{q} \overbrace{\sum_{j=0}^{q/r-1}}^{\text{neglecting small round off errors}} \exp \left(\frac{2\pi i(jr+l)c}{q} \right).$$

Upon measurement

$$\text{Prob}(c) = \frac{r}{q^2} \left| \sum_{j=0}^{q/r-1} \exp \left(\frac{2\pi i j (rc \bmod q)}{q} \right) \right|^2.$$

In the previous simpler case constructive interference occurred precisely for c satisfying $rc \bmod q = 0$.

In the above expression we look for constructive interference by considering c such that $rc \bmod q$ is suitable small. Then the added terms will all be bunched on one side of the unit circle. In fact, if

$$-r/2 \leq rc \bmod q \leq r/2,$$

the terms in $\text{Prob}(c)$ will all be spread around at most a semicircle.

There are precisely r values of $c \bmod q$ satisfying the above equation!

To see this consider the multiples of q

$$0, q, 2q, \dots, rq,$$

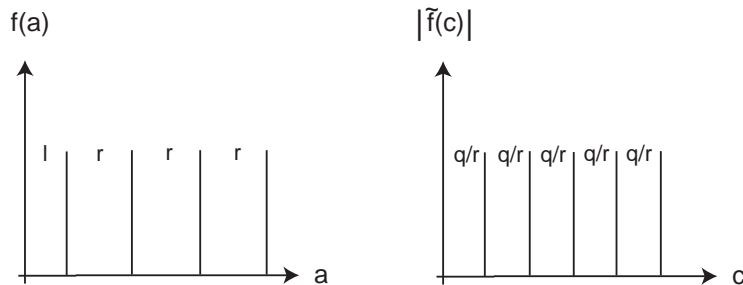
and the multiples cr of r

$$0, r, 2r, \dots, qr,$$

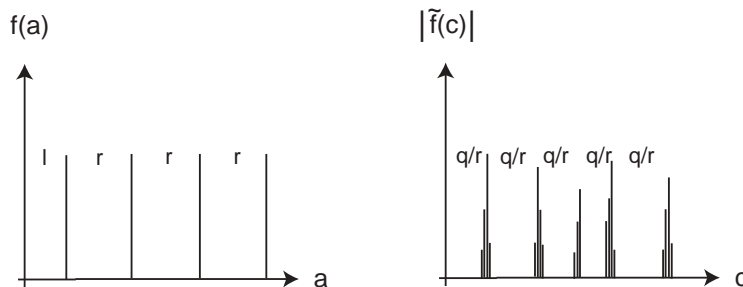
marked on the same line.

The multiples of r are spaced r apart, so, for each of the r multiples of q , there will be exactly one associated multiple of r within distance $\pm r/2$. This gives the r solutions.

$r|q$



$r \nmid q$



Estimate $\text{Prob}(c)$ for c satisfying

$$-r/2 \leq rc \bmod q \leq r/2.$$

Write $\theta_c = 2\pi(rc \bmod q)/q$ then,

$\text{Prob}(c) =$ geometric series of $e^{i\theta_c}$.

By viewing the terms of the geometric series as vectors, we see that the total distance from

the origin decreases as θ_c increases, hence

$$\text{Prob}(c) \geq \text{Prob}(c \text{ with the largest allowed } \theta_c).$$

Largest allowed θ_c is $\pi r/q$, thus

$$\begin{aligned} \text{Prob}(c) &\geq \frac{r}{q^2} \left| \sum_{j=0}^{q/r-1} \exp\left(ij \frac{\pi r}{q}\right) \right|^2 = \frac{r}{q^2} \left| 1 + \sum_{j=1}^{q/r-1} \exp\left(ij \frac{\pi r}{q}\right) \right|^2 \\ &= \frac{r}{q^2} \left| 1 + e^{i \frac{\pi r}{q}} \frac{e^{i(\frac{q}{r}-1) \frac{\pi r}{q}} - 1}{e^{i \frac{\pi r}{q}} - 1} \right|^2 \\ &= \frac{r}{q^2} \frac{1}{\sin^2 \frac{\pi r}{2q}} \approx \frac{4}{\pi^2} \frac{1}{r} \quad \text{as } \frac{r}{q} \text{ is small.} \end{aligned}$$

Since there are r such c 's, the probability of seeing a c value satisfying $-r/2 \leq rc \bmod q \leq r/2$ is greater than $4\pi^2$.

Finally we wish to extract the value of r given the value of c . Note that:

$$-\frac{r}{2} \leq rc \bmod q \leq \frac{r}{2} \implies |rc - c'q| \leq \frac{r}{2}$$

for some $0 \leq c' \leq r-1$. The r different values of c' are associated with the r possible values of c , so

$$\text{Prob}(c') \geq \frac{4}{\pi^2} \frac{1}{r}.$$

Furthermore

$$\left| \frac{c}{q} - \frac{c'}{r} \right| \leq \frac{1}{2q}.$$

Here c and q are known, and $r \leq N$, $q \geq N^2$, thus, there is exactly one fraction c'/r with denominator at most N .

It may be found efficiently using the continued fraction expansion of c/q as one of its convergents!

Continued Fractions

Definition:

$$[a_0, \dots, a_N] \equiv a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\dots + \frac{1}{a_N}}}}.$$

Definition: n th convergent to $[a_0, \dots, a_N]$ as $[a_0, \dots, a_n]$ for $0 \leq n \leq N$.

If we write the n th convergent as P_n/Q_n , then

$$\left. \begin{array}{l} P_0 = a_0, \quad P_1 = a_1 a_0 + 1, \quad P_n = a_n P_{n-1} + P_{n-2} \\ Q_0 = 1, \quad Q_1 = a_1, \quad Q_n = a_n Q_{n-1} + Q_{n-2} \end{array} \right\} \begin{array}{l} \text{recurrence} \\ \text{relations} \end{array}$$

Any (positive) rational number x can be represented by a continued fraction:

let $a_0 = \lfloor x \rfloor$, $x = a_0 + \xi_0$, $0 \leq \xi_0 \leq 1$

if $\xi_0 \neq 0 \implies$

let $a_1 = \lfloor 1/\xi_0 \rfloor$, $1/\xi_0 = a_1 + \xi_1$, $0 \leq \xi_1 \leq 1$

if $\xi_1 \neq 0 \dots$

★ Theorem

Suppose P/Q is any rational number satisfying

$$\left| \frac{P}{Q} - x \right| < \frac{1}{2Q^2},$$

then P/Q is a convergent of the continued fraction of x .

This theorem can be straightforwardly applied to find the fraction c'/r , therefore, if $\gcd(c', r) = 1$, we get the value of r !

There are $\phi(r)$ such coprime values of c' .

$\phi \rightarrow$ Euler's Phi function

$\phi(r)$ = number of integers less than r which are coprime to r .

Thus:

$$\text{Prob}(c' \text{ coprime to } r) \geq \frac{4}{\pi^2} \frac{\phi(r)}{r}.$$

For large r ,

$$\frac{\phi(r)}{r} > \frac{1}{\log r} > \frac{1}{\log N},$$

so with probability better than $4/(\pi^2 \log N)$ we obtain r and hence a factor of N !

By repeating this probabilistic algorithm $O(\log N)$ times we get an efficient factoring algorithm.

★ Factoring a 200 digit number on a quantum computer by means of the Shor's algorithm would require less than 1 hour!

CLASSIFICATION OF ALGORITHMS

1. Brute-force algorithms

This amounts to solving a problem by directly applying its crude formulation

Example: $a^n = a \cdot a \cdot \dots \cdot a$ n times.

2. Divide and conquer algorithms

The original problem is partitioned into a number of smaller sub-problems. Once these are solved, their solutions are combined to get the solution of the bigger problem.

Example: $a^n = a^{\lfloor n/2 \rfloor} \cdot a^{\lfloor n/2 \rfloor} \cdot a^{n-2\lfloor n/2 \rfloor}$.

3. Decrease and conquer algorithms

The original problem is reduced to a smaller one, which is solved by recursion and the solution is then applied to find the solution of the original problem.

Example: $a^n = a^{n-1} \cdot a$.

4. Transform and conquer algorithms

The original problem is transformed into another equivalent problem more manageable.

Example: a^n computed by means of the binary form of n .

Technique	Classical Algorithms	Quantum Algorithms
Brute-force	Searching the largest	Grover, Deutsch-Jozsa
Divide and conquer	Quicksort	—
Decrease and conquer	Euclid	—
Transform and conquer	Gaussian elimination	Shor

★ The absence of quantum algorithms based on divide and conquer, and decrease and conquer could be ascribed to quantum parallelism and entanglement which make the naive implementation of such strategies quite unnatural.