Shor's Algorithm

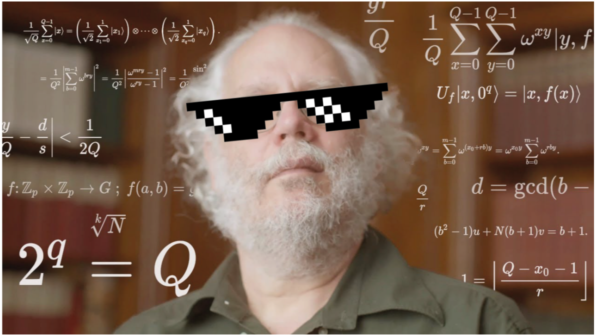
Pagonis Alexandros

QSilver (QGreece)

April 28, 2022

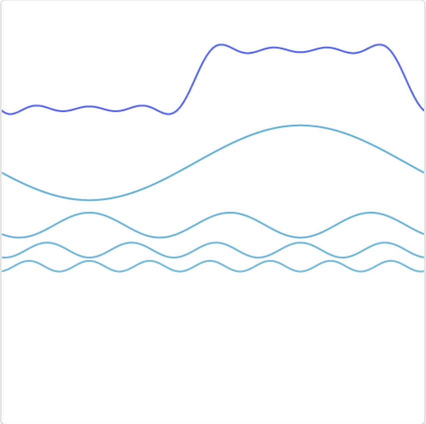
Overview

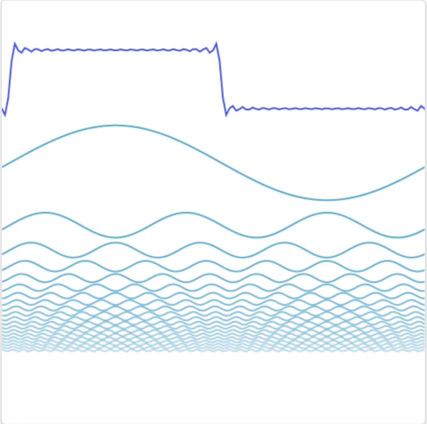
Factorize? → **QFT** → **QPE** → **Order Finding** → **Shor**



Look what I found...

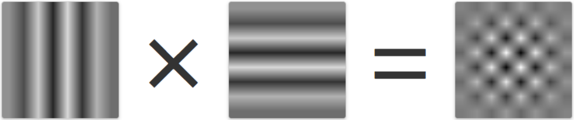
Fourier Transform

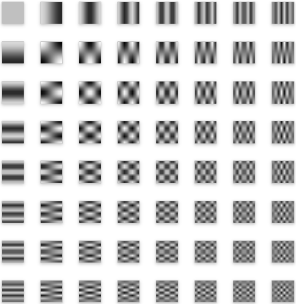




Fourier picks up frequencies!

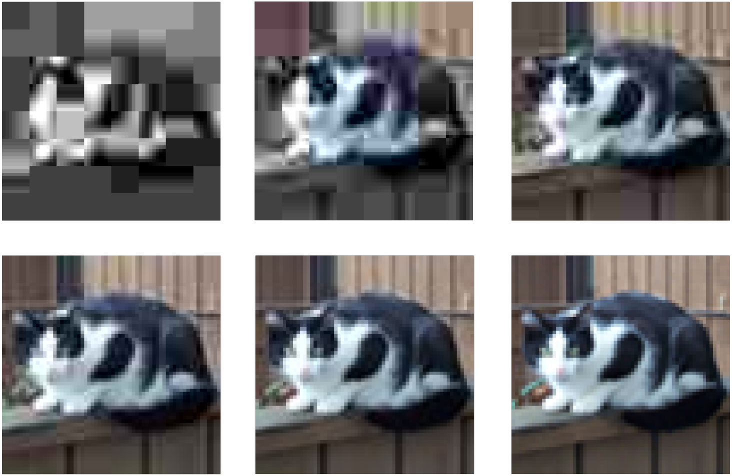
2D Fourier Transform





What if I add vertical and horizontal "waves"?

Fourier seems useful



I can approximate any picture with these blocks!

Review: (Quantum) Fourier Transform

Vector: $x = (x_0\ x_1\ \dots\ x_{N-1})^T$
Discrete Fourier Transform:

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

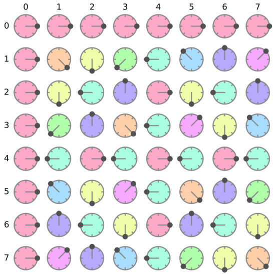
$$\frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2N-2} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3N-3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots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

with

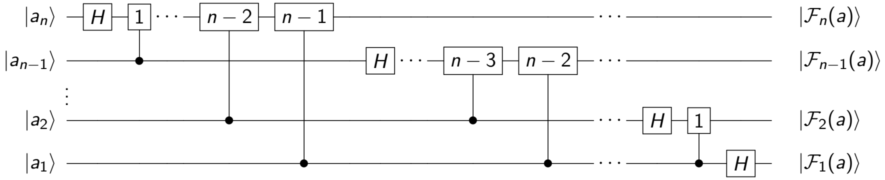
$$\omega = e^{\frac{2\pi i}{N}}$$

Quantum Fourier Transform:
(on base state)

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



QFT Circuit



where $\boxed{k}$ is the gate corresponding to $P(\frac{\pi}{2^k}) = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{2^k}} \end{pmatrix}$.

Review: (Quantum) Phase Estimation Algorithm

I want to "measure" ϕ from this circuit:

$$|\psi\rangle \; -\boxed{U}- \qquad e^{i\phi}|\psi\rangle$$

I can fetch a phase:

$$\begin{array}{ccc} \frac{|0\rangle+|1\rangle}{\sqrt{2}} & \text{---}\bullet\text{---} & \frac{|0\rangle+e^{i\phi}|1\rangle}{\sqrt{2}} \\ |\psi\rangle & -\boxed{U}- & |\psi\rangle \end{array}$$

If I could make the state:

$$|F_{|\phi\rangle}\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi i\phi k}{N}} |k\rangle$$

I could "print" it by reversing QFT:

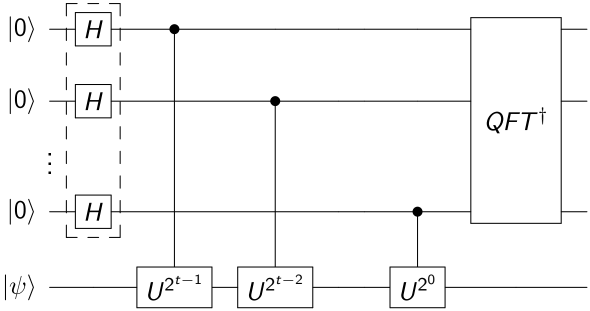
$$\mathrm{QFT}^\dagger |F_{|\phi\rangle}\rangle = |\phi\rangle$$

Review: (Quantum) Phase Estimation Circuit

I can construct the desired state:

$$|F_{|\phi\rangle}\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi i \phi k}{N}} |\phi\rangle$$

Whole QPE seems like this:



Review: Find Order by finding period

- Order r of x: $x^r = 1 \pmod{N}$
- Well I know a trivial one: $x^0 = 1 \pmod{N}$
- So all "orders" are multiples of r
- Turns out the function $f(r) = x^r \pmod{N}$ is periodic
- Find r simply means: find the period of $f(r)$!
- Idea: Fetch period in QFT. Use QPE to print it so we can read it
- I just need the right gate. One that spits out the period!!!

Review: (Quantum) Order Finding Algorithm (1)

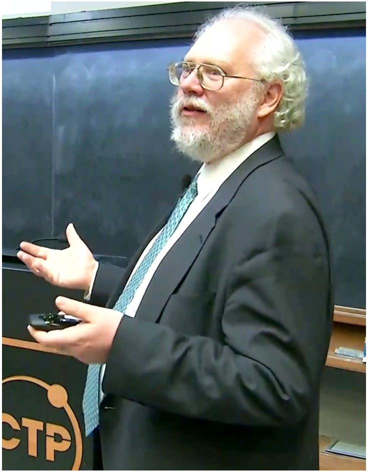
- Order r of x: $x^r = 1 \pmod N$
- Operator: $U_x \left|y\right\rangle = \left|xy \pmod N\right\rangle$
- U_x eigenstates: $\left|u_s\right\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-\frac{2\pi i s k}{r}} \left|x^k \pmod N\right\rangle$
- Indeed: $U_x \left|u_s\right\rangle = e^{\frac{2\pi i s}{r}} \left|u_s\right\rangle$
- Eigenstates contain r, but you don't know it yet you idiot!
- Hack: I can't construct one of them. But all of them add up to 1: $\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \left|u_s\right\rangle = \left|1\right\rangle$

Review: (Quantum) Order Finding Algorithm (2)

- Initial State: $\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |0\rangle^{\otimes t} |u_s\rangle = |0\rangle^{\otimes t} |1\rangle$
- *QFT* (superposition): $\frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle |1\rangle$
- Apply U_x: $\frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle |x^j \pmod N\rangle \approx \frac{1}{\sqrt{r2^t}} \sum_{s=0}^{r-1} \sum_{j=0}^{2^t-1} e^{2\pi i sj/r} |j\rangle |u_s\rangle$
- Apply $QFT^\dagger$: $\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |\sim s/r\rangle |u_s\rangle$
- Measure 1st register: $\sim s/r$
- Continued Fractions: r

Factorization (and why do we care?)

It is extremely difficult to factorize a composite number N into its prime factors:
$N = p_1^{k_1} \dots p_n^{k_n}$.
The less factors the worse.
Worst case scenario $\boxed{N = pq}$, where p, q are two big prime numbers.
Most of our encryption schemes count on this difficulty (such as RSA).



Algorithm Steps

Goal: Factorize N=pq. Meaning: find p,q.

- Step 1: Pick a random number a in range $\{2...N-1\}$
- Step 2: Unless we accidentally chose p or q: $gcd(a, N) = 1$
- Step 3: Find order of a. Smallest r such that: $a^r \equiv 1 \pmod{N}$
- Check 1: If r is odd, go back to step 1
- Check 2: If $a^{\frac{r}{2}} \equiv N - 1 \equiv -1 \pmod{N}$, go back to step 1
- Finally: $p, q = gcd(a^{\frac{r}{2}} \pm 1, N)$

Warm up (1)

Can I factorize this: $x^2 - 1$?

Warm up (2)

Can I factorize this: $x^2 - 1$?

Yes I can: $\boxed{x^2 - 1 = (x-1)(x+1)}$

Warm up (3)

Can I factorize this: $a^r - 1$?

Warm up (4)

Can I factorize this: $a^r - 1$?

Yes I can: $\boxed{a^r - 1 = (a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1)}$

Why does it work? (1)

$$
\begin{aligned}
a^r &\equiv 1 \pmod{N} \implies \\
a^r - 1 &\equiv 0 \pmod{N} \overset{\text{Check 1}}{\implies} \\
(a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1) &\equiv 0 \pmod{N} \implies \\
N &| (a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1)
\end{aligned}
$$

Why does it work? (2)

Remember Check 2:

$$a^{\frac{r}{2}} \not\equiv -1 \pmod N \implies$$
$$a^{\frac{r}{2}} + 1 \not\equiv 0 \pmod N \implies$$
$$N \not| (a^{\frac{r}{2}} + 1)$$

But:

$$N | (a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1)$$

Why does it work? (3)

Eq1:

$$pq \nmid (a^{\frac{r}{2}} + 1)$$

Eq2:

$$pq \mid (a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1)$$

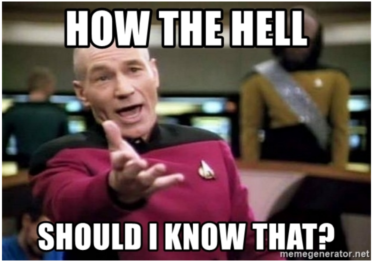
I found a divisor (common factor):

$$p = (a^{\frac{r}{2}} + 1)$$

And thus I found the other one too:

$$q = (a^{\frac{r}{2}} - 1)$$

What about the 2 checks?



- r has to be even so that $(a^{\frac{r}{2}} + 1)$ and $(a^{\frac{r}{2}} - 1)$ are integers.
- It must also: $a^{\frac{r}{2}} \equiv -1 \pmod{N}$

Theorem

Theorem (What a useful theorem)

Suppose $N = {p_1}^{l_1} \dots {p_m}^{l_m}$ is the prime factorization of an odd composite positive integer. Let x be an integer uniformly chosen at random, such that $0 \leq x \leq N-1$ and x is co-prime to N. Let r be the order of $x \pmod{N}$. In such a case,

$$P(r \text{ is even and } x^{r/2} \neq -1 \pmod{N}) > 1 - \frac{1}{2^{m-1}}.$$

In the worst case $N = pq$ so $P(\text{checks}) > 1 - \frac{1}{2^{2-1}} = 50\%$.

General Case: many factors

- Suppose $N = {p_1}^{l_1} \dots {p_m}^{l_m}$
- Step 1: Pick a random number a in range $\{2 ... N-1\}$
- Step 2: Unless we accidentally chose a factor: $gcd(a, N) = 1$
- Step 3: Find order of a. Smallest r such that: $a^r \equiv 1 \pmod N$
- Check 1: If r is odd, go back to step 1
- Check 2: If $a^{\frac{r}{2}} \equiv N - 1 \equiv -1 \pmod N$, go back to step 1
- One of two values is a factor: $f = gcd(a^{\frac{r}{2}} \pm 1, N)$
- Continue with what's left: $N_{new} = N/f$

Questions?

The End