SHOR’S ALGORITHM

Given a positive integer, find its factors. There exists a fast classical algorithm for detecting whether the number is prime. If so, problem solved. There exists a fast classical algorithm for detecting whether the number is a power (ie. of the form $a^b$ for $b > 1$)

$$x = a^b$$

$$\log x = b \times \log a$$ so $b$ cannot be bigger than $\log x$

Binary search: : If you can find a factor, you can find all other factors as well using the same method repeatedly.

Assume you are given a number pq where p and q are primes.

We want to factor a large integer N.

- FACTORING is reduced to finding a non-trivial square root of 1 modulo $N \Rightarrow y^2 \equiv (mod 15) y \in (1, 2, 3, \ldots, N - 1)$

if $N=15$

trivial square roots:

$1^2 = 1(mod 15)$  $(-1)^2 = 1(mod 15)$

$14^2 = 1(mod 15)$

example non-trivial root:

$4^2 = 1(mod 15)$

if $y$ is a non-trivial square root of 1 mod N, then N divides $y^2 - 1 \equiv (y + 1)(y - 1)$, but N does not divide neither $(y-1)$ nor $(y+1)$ so this
means that \( \gcd(y - 1, N) > 1 \) because if \( y-1 \) and \( N \) were relatively prime, then since \( N \) divides \((y - 1)(y + 1)\), it would have to divide \((y + 1)\).

- Finding such a root is reduced to computing the order of a random integer modulo \( N \).

Pick a random number \( X \mod N \), order of \( X \mod N \) is the smallest number \( r \) such that \( x^r \equiv 1 \mod N \).

Note: Until this step everything can be solved by classical machines in a fast manner.

- The order of an integer is precisely the period of a particular periodic superposition

\[
\begin{array}{cccccccc}
2 & 4 & 8 & 1 & 2 & 4 & 8 & 1 \\
\end{array}
\]

Classical FFT’s input is an \( M \)-dimensional complex valued vector \( \alpha \) (Where \( M \) is a power of 2) and output is a \( M \)-dimensional complex valued vector \( \beta \).

\[
\begin{bmatrix}
\beta_0 \\
\beta_1 \\
\beta_2 \\
\vdots \\
\beta_{M-1}
\end{bmatrix} = \frac{1}{\sqrt{M}}
\begin{bmatrix}
1 & 1 & 1 & \ldots & 1 \\
1 & w & w^2 & \ldots & w^{m-1} \\
1 & w^2 & w^4 & \ldots & w^{2(m-1)} \\
1 & w^3 & w^6 & \ldots & w^{3(m-1)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & w^{m-1} & w^{2(m-1)} & \ldots & w^{(m-1)(m-1)}
\end{bmatrix}
\begin{bmatrix}
\alpha_0 \\
\alpha_1 \\
\alpha_2 \\
\vdots \\
\alpha_{M-1}
\end{bmatrix}
\]

Where \( W \) is a complex \( M \)th root of unity.

FFT runs in \( O(M \log M) \) steps.

- And, periods of superposition can be found by the quantum FFT.

Since whole \( \beta \) cannot be accessed because of the quantum reality this is called “Quantum Fourier Sampling”.

Input: A superposition of \( M = \log M \) qubits \( |\alpha> = \sum_{i=0}^{M-1} \alpha_i |j> \).

Method: We’ll use \( O(m^2) \) quantum operations to obtain superposition \( |\beta> = \sum_{i=0}^{M-1} \beta_i |j> \).

Output: A random \( m \)-bit number \( j \) (ie. \( 0 \leq j \leq M - 1 \)), from the probability distribution \( P_i[j] = |B_i|^2 \).
Suppose the input to quantum Fourier sampling is periodic with the period \( k \), for some \( k \) that divides \( M \). Then the output will be a multiple of \( M/k \), and it is equally likely to be any of the \( k \) multiples of \( M/k \).

\[
|\alpha> = \begin{bmatrix}
\alpha_0 \\
\alpha_1 \\
\alpha_2 \\
\vdots \\
\alpha_{M-1}
\end{bmatrix}
\]
is such that \( \alpha_j = \alpha_i \) whenever \( i = j \pmod{k} \) where \( k \) is a particular integer that divides \( M \). So there are \( M/k \) repetitions of same sequence \( (\alpha_0, \alpha_1, \ldots, \alpha_{M-1}) \) of length \( k \). AND SUPPOSE EXACTLY ONE OF THE \( k \) NUMBERS IS NONZERO say \( \alpha_j \).

Suppose the vector \( (\alpha_0, \alpha_1, \ldots, \alpha_{M-1})^T \) is periodic with period \( k \) with no offset. Thus \( |\alpha> = \sum_{j=0}^{M/k-1} \sqrt{\frac{k}{M}} |jk> \) (non-zero terms are \( \alpha_0, \alpha_1, \ldots \) )

Claim: \( |\beta> = \frac{1}{\sqrt{k}} \sum_{j=0}^{k-1} |\frac{M}{k}j> \)

In the input vector, the coefficient of \( \alpha_l \) is \( \frac{k}{M} \) if \( k \) divides \( l \), and zero otherwise. The \( j \)th coefficient of \( |\beta> \) is \( |\beta> = \frac{1}{\sqrt{M}} \sum_{l=0}^{M-1} w^{jl} \alpha_l =
\sqrt{k} \sum_{i=0}^{Wk-1} w^{ji} \).

So this sum is the geometric series \( 1 + w^{jk} + w^{2jk} + \ldots \) containing \( \frac{M}{k} \) terms and with ratio \( w^{jk} \). There are two cases. If the ratio is exactly 1, which happens if \( jk = 0 \pmod{M} \), then the sum of the series is just the number of terms.

If the ratio isn’t 1, apply the usual formula for geometric series to find that sum.

So \( \beta_j = \frac{1}{\sqrt{k}} \) is \( M \) divides \( jk \), and is zero otherwise.

Also work (with little mod function) for the case where the offset isn’t zero.

Suppose s independent samples are drawn uniformly from 0, \( \frac{M}{k} \), \( \frac{2M}{k} \), \( \frac{3M}{k} \), \ldots, \( \frac{(k-1)M}{k} \).

Then with probability at least \( 1 - \frac{k}{2} \), the greatest common divisor of these samples is \( M/k \).

Proof: The only way this can fail is if all the samples are multiples of \( j \). \( M/k \), for some \( j > 1 \). So fix any integer \( j \geq 2 \).

The chance that particular sample is a multiple of \( \frac{M}{k} \) is at most \( \frac{1}{j} \leq \frac{1}{2} \),
so the chance that all samples are multiples of $\frac{jM}{k}$ is at most $\frac{1}{2^s}$. The probability that this bad thing will happen for some $j \leq k$ is at most $k\frac{1}{2^s}$, since there are $k$ candidates for number $j$.

How does the classical FFT work?

Divide and Conquer

From input $(\alpha_0, \alpha_1, \ldots, \alpha_{M-1})^T$ to output $(\beta_0, \beta_1, \ldots, \beta_{M-1})^T$.

In the quantum version, the input is now encoded in the $2^M$ amplitudes of $m = \log_2 M$ qubits. So the decomposition of the inputs to evens and odds is determined by the least significant qubit.

We will design a quantum circuit (subroutine) $QFT_M$. $QFT_{M/2}$ will be applied to the remaining $m - 1$ qubits.

Sidenote:

For the 6 qubit system:
\[ \text{Gate with 2 qubits} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & W^2 \end{bmatrix} \]