Quantum Software Development

Lecture 10: Shor’s Factorization Algorithm

April 10, 2024

Tufts UNIVERSITY | School of Engineering

MITRE | SOLVING PROBLEMS FOR A SAFER WORLD®

Shor’s Factorization Algorithm

Integer factorization is thought to be intractable on a classical computer.

Multiplication is easy.

$$53 \times 71 = ?$$

$$\begin{array}{r} 53 \\ \times\ 71 \\ \hline 53 \\ +\ 371 \\ \hline =\ 3763 \end{array}$$

$$O\left((\log N)^2\right)$$

Factorization is hard.

$$3763 = ? \times ?$$

$$3763 \bmod 2 \neq 0$$

$$3763 \bmod 3 \neq 0$$

$$\vdots$$

$$3763 \bmod 53 = 0$$

$$O\left(\sqrt{N}\right)$$

MITRE

Shor’s algorithm works by reducing the problem to finding the period of a modular exponentiation function.

1. Guess a number g between 1 and the number to factor N.
2. Check if g and N share a common factor; if they do, we’re done!
3. Find the period p of the function $f(x) = g^x \bmod N$.
4. If p is odd or $\left(g^{\frac{p}{2}} + 1\right) \bmod N = 0$, go back to step 1.
5. Otherwise, $g^{\frac{p}{2}} + 1$ or $g^{\frac{p}{2}} - 1$ are guaranteed to share a common factor with N.

Use Euclid’s GCD algorithm, $O(\log N)$

!?

Modular exponentiation is periodic if the base and modulus are relatively prime.

x	$5^x \bmod 21$
0	1
1	5
2	4
3	20
4	16
5	17
6	1
7	5
⋮	⋮

Cycle repeats at $x = 6$

Finding the period p of $f(x) = g^x \bmod N$ gives $g^p \bmod N = 1$.

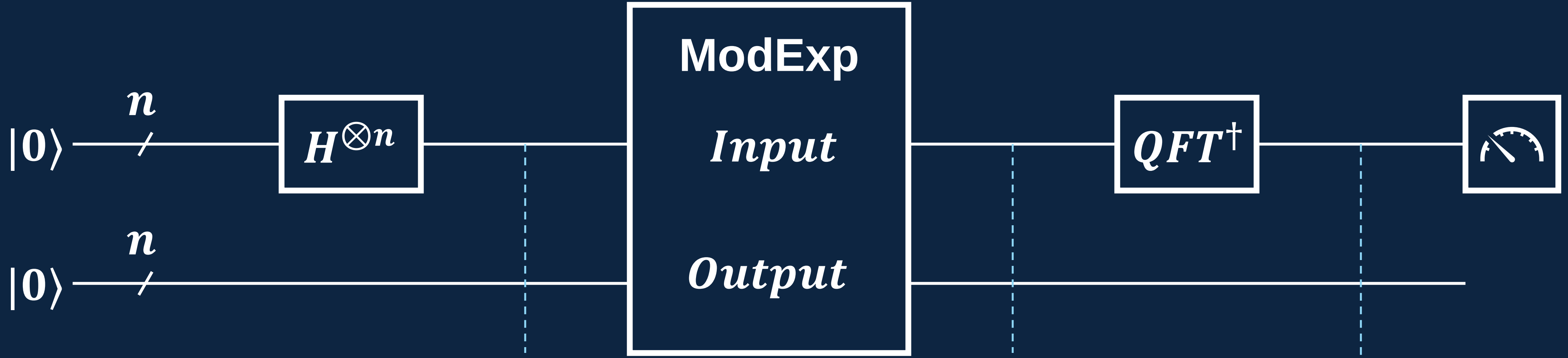
This implies $g^p - 1 = mN$, for some integer m.

Factoring using the difference of squares gives $\left(g^{\frac{p}{2}} + 1\right)\left(g^{\frac{p}{2}} - 1\right) = mN$.

Assuming p is even and $g^{\frac{p}{2}} + 1$ is not a multiple of N, one of the terms must share a common factor with N.

MITRE

How might a quantum computer be used to find the period of the modular exponentiation function?



$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, 0\rangle \qquad\qquad \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, f(x)\rangle$$

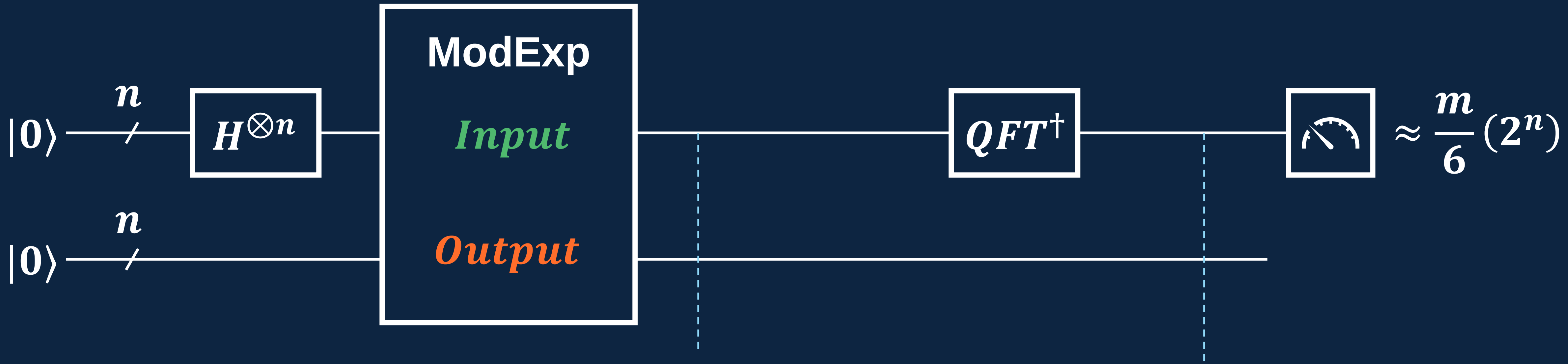
Inputs having the same output are grouped; each group of input values has period p.

Each input group produces frequency bins at* multiples of $\frac{1}{p}$

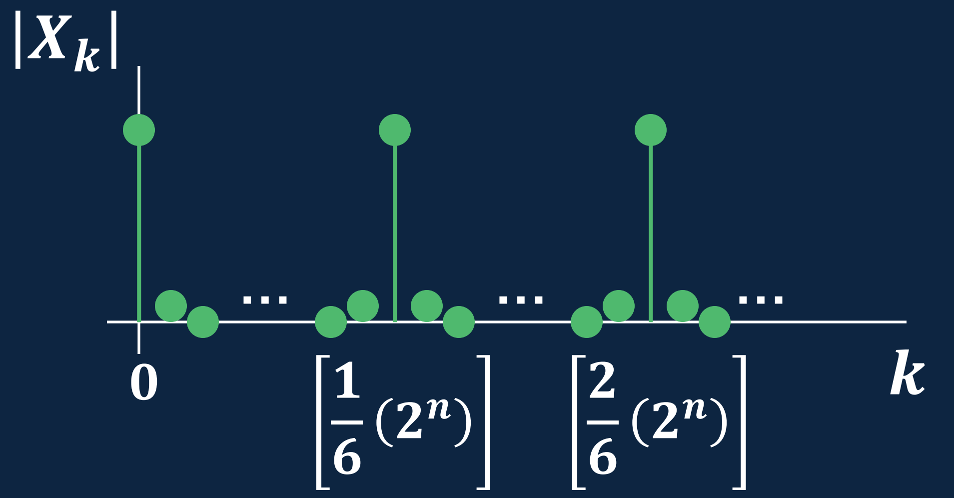
*approximately

MITRE

Example: $f(x) = 5^x \bmod 21$



$$\frac{1}{\sqrt{2^n}}\left(\begin{array}{c}
(|0\rangle + |6\rangle + \cdots) \otimes |1\rangle + \\
(|1\rangle + |7\rangle + \cdots) \otimes |5\rangle + \\
(|2\rangle + |8\rangle + \cdots) \otimes |4\rangle + \\
(|3\rangle + |9\rangle + \cdots) \otimes |20\rangle + \\
(|4\rangle + |10\rangle + \cdots) \otimes |16\rangle + \\
(|5\rangle + |11\rangle + \cdots) \otimes |17\rangle
\end{array}\right)$$



Period-Finding Subroutine

1. **Set up two registers of length n such that $N^2 \leq 2^n < 2N^2$.**
   (Alternatively, $n = \lceil 2 \log_2 N \rceil$.)

2. **Put the input register into a uniform superposition.**

3. **Apply modular exponentiation as a quantum operation.**

4. **Apply the inverse QFT to the input register.**

5. **Measure the input register.**

6. **Use continued fraction expansion to approximate p.**
   If this fails, go back to step 1.

Modular exponentiation is the bottleneck. It is roughly as hard as multiplication, $O\left((\log N)^2\right)$.

MITRE

Shor's Factorization Algorithm

1. Pick some integer g such that $1 < g < N$, where N is the number to factor
2. Compute $GCD(g, N)$; if the result is > 1, it's a factor of N and we're done
3. Find the period p of the function $f(x) = g^x \bmod N$, giving $g^p \bmod N = 1$
   A. Set up two registers $|I, O\rangle = |0^{\otimes n}, 0^{\otimes n}\rangle$, where $N^2 \leq 2^n < 2N^2$
   B. Apply $H^{\otimes n}$ to put $|I\rangle$ into a uniform superposition
   C. Apply $f(x)$ as a quantum operation that maps $|x, 0\rangle \rightarrow |x, f(x)\rangle$
   D. Apply $QFT^\dagger$ to $|I\rangle$
   E. Measure $|I\rangle$ and obtain some value X
   F. Use continued fraction expansion on $\frac{X}{2^n}$ to find candidates for p
4. If p is odd or $\left(g^{\frac{p}{2}} + 1\right) \bmod N = 0$, fail
5. Compute $GCD\left(g^{\frac{p}{2}} \pm 1, N\right)$; guaranteed to get at least one factor of N

repeat until success

repeat until success

MITRE

The Euclidean algorithm can find the greatest common divisor of two numbers efficiently.

$$
\begin{aligned}
&\boldsymbol{GCD(A,B)}:\\
&\quad \boldsymbol{if\ B = 0,\ return\ A};\\
&\quad \boldsymbol{return\ GCD(B, A \bmod B)};
\end{aligned}
$$

$$\boldsymbol{GCD(91,49)}$$

$$\downarrow \quad 91 \bmod 49 = 42$$

$$\boldsymbol{GCD(49,42)}$$

$$\downarrow \quad 49 \bmod 42 = 7$$

$$\boldsymbol{GCD(42,7)}$$

$$\downarrow \quad 42 \bmod 7 = 0$$

$$\boldsymbol{GCD(7,0)}$$

$$\downarrow$$

$$\boxed{\boldsymbol{7}}$$

In Shor's algorithm, if we find an integer A such that $1 < GCD(A,N) < N$, then $GCD(A,N)$ is a factor of N!

Modular exponentiation can be performed efficiently using the binary substitution method.

How do we compute $f(x) = g^x \bmod N$?

- **Express x in little-endian binary notation:**

$$x = x_0 2^0 + x_1 2^1 + \cdots + x_{n-1} 2^{n-1}$$

- **Break g^x up into n terms:**

$$g^{x_0 2^0 + x_1 2^1 + \cdots + x_{n-1} 2^{n-1}} = g^{x_0 2^0} \cdot g^{x_1 2^1} \cdot \ldots \cdot g^{x_{n-1} 2^{n-1}}$$

- **Compute each term one-at-a-time under $\bmod N$:**

$$g^x \bmod N =$$
$$\left(g^{x_0 2^0} \bmod N\right) \cdot \left(g^{x_1 2^1} \bmod N\right) \cdot \ldots \cdot \left(g^{x_{n-1} 2^{n-1}} \bmod N\right)$$

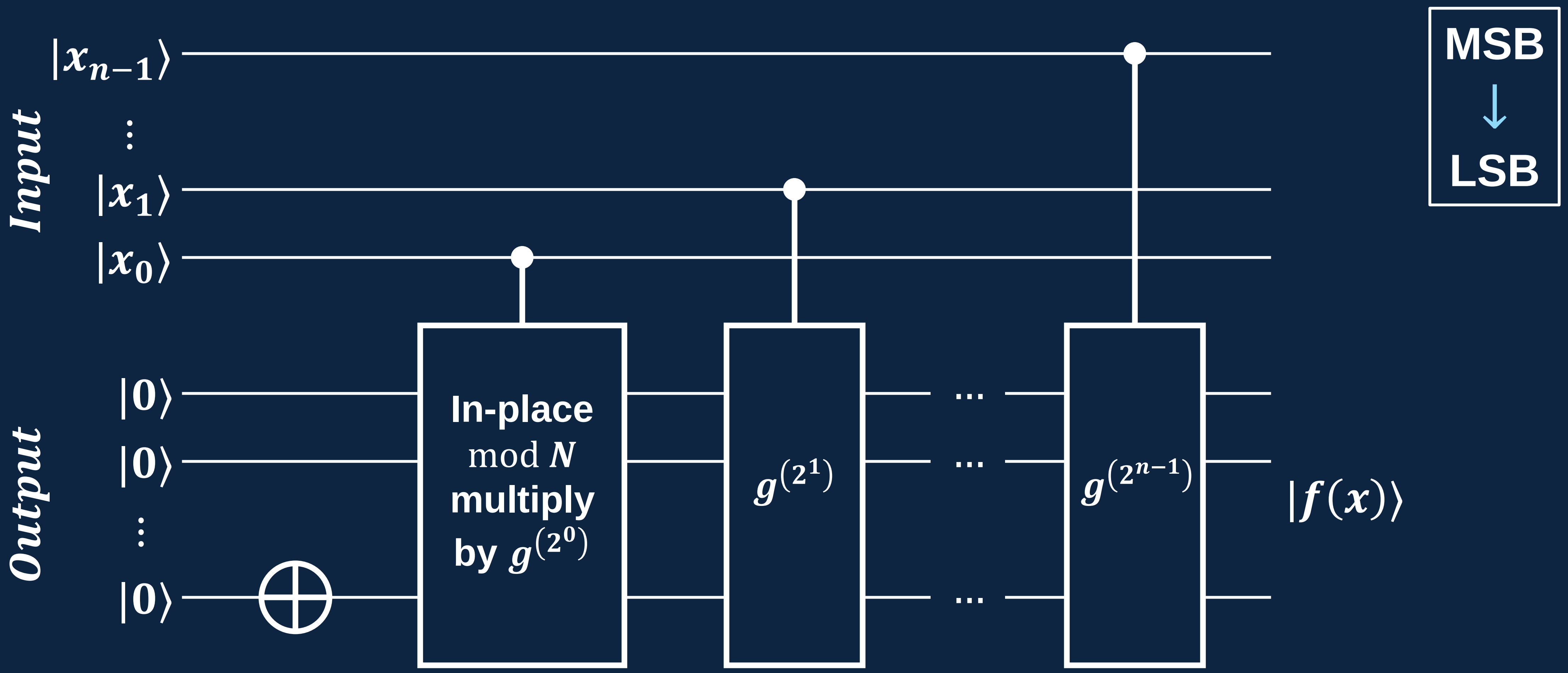
Modular Exponentiation Procedure

1. Initialize $f_{temp} = 1$
2. Iterate over the bits in x starting with the LSB; if $x_i = 1$ do:
   A. Multiply f_{temp} by $g^{2^i} \bmod N$
   B. Set f_{temp} to $f_{temp} \bmod N$
3. Now, $f_{temp} = f(x)$

$\boxed{f_{temp} \text{ never exceeds } N^2}$

MITRE

The binary substitution method is straightforward to implement as a quantum operation.



Any rational number can be represented as a continued fraction.

$$\frac{P}{Q} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}}$$

$$\frac{13}{16} = 0 + \cfrac{1}{1 + \cfrac{1}{4 + \cfrac{1}{3}}}$$

Continued Fraction Expansion Procedure

1. **Initialize $P_i = P$, $Q_i = Q$, $i = 0$**
2. **Perform integer division $P_i \div Q_i$; the quotient is a_i and the remainder is r_i**
3. **If $r_i = 0$, we're done**
4. **Repeat with $P_{i+1} = Q_i$, $Q_{i+1} = r_i$, $i = i + 1$**

i	P_i	Q_i	a_i	r_i
0	13	16	0	13
1	16	13	1	3
2	13	3	4	1
3	3	1	3	0

MITRE

Continued fraction expansion can approximate the period p based on the measured inverse QFT result.

In Shor's algorithm, measuring a value $|X\rangle$ implies a frequency bin of $\frac{X}{2^n}$, which is close to a multiple of $\frac{1}{p}$.

This (surprisingly) works:

- Do continued fraction expansion with $P = X$, $Q = 2^n$.
- Check the approx. value "so far" after each iteration:

$$v_i = \frac{m_i}{d_i} = a_0 + \frac{1}{a_1 + \frac{1}{\dots + \frac{1}{a_i}}}$$

$$m_i = a_i \cdot m_{i-1} + m_{i-2}$$
$$d_i = a_i \cdot d_{i-1} + d_{i-2}$$

- Stop when $d_i \geq N$ and take d_{i-1} as a candidate for p.

$$\frac{13}{16} = 0 + \frac{1}{1 + \frac{1}{4 + \frac{1}{3}}}$$

i	a_i	m_i	d_i	v_i
0	0	0	1	$\frac{0}{1}$
1	1	1	1	$\frac{1}{1}$
2	4	4	5	$\frac{4}{5}$
3	3	13	16	$\frac{13}{16}$

MITRE

Example: Factor 143 using Shor's algorithm.

1. Pick $g = 10$
2. Compute $GCD(10, 143) = 1$ … 10 and 143 are coprime
3. Find the period p of the function $f(x) = 10^x \bmod 143$, giving $10^p \bmod 143 = 1$
   - $n = \lceil 2 \log_2 143 \rceil = 15$
   - Likely to measure some X such that $\frac{X}{2^{15}} \approx \frac{m}{p}$
   - Suppose we measure $X = 27307$
   - Do continued fraction expansion on $\frac{27307}{32768}$
   - $\frac{27307}{32768} \approx \frac{5}{6}$, so 6 is a candidate for p
   - Check $10^6 \bmod 143 = 1$ ✓
4. Check $\left(10^{\frac{6}{2}} + 1\right) \bmod 143 = 0$ … $1001 = 143 \cdot 7$ so try again …

x	$10^x (\mathrm{mod}\ 143)$
0	1
1	10
2	100
3	142
4	133
5	43
6	1
⋮	⋮

MITRE

Example: Factor 143 using Shor's algorithm (take 2).

1. Pick $g = 12$
2. Compute $GCD(12, 143) = 1$ … 12 and 143 are coprime
3. Find the period p of the function $f(x) = 12^x \bmod 143$, giving $12^p \bmod 143 = 1$
   - $n = \lceil 2\log_2 143 \rceil = 15$
   - Likely to measure some X such that $\frac{X}{2^{15}} \approx \frac{m}{p}$
   - Suppose we measure $X = 16384$
   - $\frac{16384}{32768} = \frac{1}{2}$, so 2 is a candidate for p
   - Check $12^2 \bmod 143 = 1$ ✓
4. Check $\left(12^{\frac{2}{2}} + 1\right) \bmod 143 = 0 \ldots 13 \bmod 143 = 13$ ✓
5. Compute $GCD(13, 143) = 13$ … 13 is a factor of 143!

x	$12^x (\mathrm{mod}\ 143)$
0	1
1	12
2	1
3	12
4	⋮

MITRE