Today's outline - March 03, 2022

Today's outline - March 03, 2022

- Overview of Shor's algorithm

Today's outline - March 03, 2022

- Overview of Shor's algorithm
- Period-finding and factoring strategy

Today's outline - March 03, 2022

- Overview of Shor's algorithm
- Period-finding and factoring strategy
- Shor's factoring algorithm

Today's outline - March 03, 2022

- Overview of Shor's algorithm
- Period-finding and factoring strategy
- Shor's factoring algorithm
- The quantum core, and period extraction

Today's outline - March 03, 2022

- Overview of Shor's algorithm
- Period-finding and factoring strategy
- Shor's factoring algorithm
- The quantum core, and period extraction
- An example of Shor's algorithm

Today's outline - March 03, 2022

- Overview of Shor's algorithm
- Period-finding and factoring strategy
- Shor's factoring algorithm
- The quantum core, and period extraction
- An example of Shor's algorithm

Reading assignment: 8.3 – 8.4

Today's outline - March 03, 2022

- Overview of Shor's algorithm
- Period-finding and factoring strategy
- Shor's factoring algorithm
- The quantum core, and period extraction
- An example of Shor's algorithm

Reading assignment: 8.3 – 8.4

Homework Assignment #05:
Chapter 7:1,3,4
due Sunday, March 06, 2022

Today's outline - March 03, 2022

- Overview of Shor's algorithm
- Period-finding and factoring strategy
- Shor's factoring algorithm
- The quantum core, and period extraction
- An example of Shor's algorithm

Reading assignment: 8.3 – 8.4

Homework Assignment #05:
Chapter 7:1,3,4
due Sunday, March 06, 2022

HW #06 will include using quirk

Today's outline - March 03, 2022

- Overview of Shor's algorithm
- Period-finding and factoring strategy
- Shor's factoring algorithm
- The quantum core, and period extraction
- An example of Shor's algorithm

Reading assignment: 8.3 – 8.4

Homework Assignment #05:
Chapter 7:1,3,4
due Sunday, March 06, 2022

HW #06 will include using quirk

Quantum circuit simulator https://algassert.com/quirk

Classical period-finding

In 1994 Shor developed an algorithm for factoring integers which coupled with the quantum Fourier transform threatened to crack the standard encryption algorithms of the time

Classical period-finding

In 1994 Shor developed an algorithm for factoring integers which coupled with the quantum Fourier transform threatened to crack the standard encryption algorithms of the time

The factoring algorithm relies finding the period of a function $f(k)$

Classical period-finding

In 1994 Shor developed an algorithm for factoring integers which coupled with the quantum Fourier transform threatened to crack the standard encryption algorithms of the time

The factoring algorithm relies finding the period of a function $f(k)$

The order of an integer $a \bmod M$ is the smallest integer r such that $r > 0$ and $a^r = 1 \bmod M$

Classical period-finding

In 1994 Shor developed an algorithm for factoring integers which coupled with the quantum Fourier transform threatened to crack the standard encryption algorithms of the time

The factoring algorithm relies finding the period of a function $f(k)$

The order of an integer $a \bmod M$ is the smallest integer r such that $r > 0$ and $a^r = 1 \bmod M$

Classical period-finding

In 1994 Shor developed an algorithm for factoring integers which coupled with the quantum Fourier transform threatened to crack the standard encryption algorithms of the time

The factoring algorithm relies finding the period of a function $f(k)$

The order of an integer $a \bmod M$ is the smallest integer r such that $r > 0$ and $a^r = 1 \bmod M$

If the two integers a and M are relatively prime (i.e. they share no prime factors) then r exists and the order of a is finite

Classical period-finding

In 1994 Shor developed an algorithm for factoring integers which coupled with the quantum Fourier transform threatened to crack the standard encryption algorithms of the time

The factoring algorithm relies finding the period of a function $f(k)$

The order of an integer $a \bmod M$ is the smallest integer r such that $r > 0$ and $a^r = 1 \bmod M$

If the two integers a and M are relatively prime (i.e. they share no prime factors) then r exists and the order of a is finite

Consider the function $f(k)$

Classical period-finding

In 1994 Shor developed an algorithm for factoring integers which coupled with the quantum Fourier transform threatened to crack the standard encryption algorithms of the time

The factoring algorithm relies finding the period of a function $f(k)$

The order of an integer $a \bmod M$ is the smallest integer r such that $r > 0$ and $a^r = 1 \bmod M$

If the two integers a and M are relatively prime (i.e. they share no prime factors) then r exists and the order of a is finite

Consider the function $f(k)$

$$f(k) = a^k \bmod M$$

Classical period-finding

In 1994 Shor developed an algorithm for factoring integers which coupled with the quantum Fourier transform threatened to crack the standard encryption algorithms of the time

The factoring algorithm relies finding the period of a function $f(k)$

The order of an integer $a \bmod M$ is the smallest integer r such that $r > 0$ and $a^r = 1 \bmod M$

If the two integers a and M are relatively prime (i.e. they share no prime factors) then r exists and the order of a is finite

Consider the function $f(k)$
Since $a^r = 1 \bmod M$ we can write

$$f(k) = a^k \bmod M$$

Classical period-finding

In 1994 Shor developed an algorithm for factoring integers which coupled with the quantum Fourier transform threatened to crack the standard encryption algorithms of the time

The factoring algorithm relies finding the period of a function $f(k)$

The order of an integer $a \bmod M$ is the smallest integer r such that $r > 0$ and $a^r = 1 \bmod M$

If the two integers a and M are relatively prime (i.e. they share no prime factors) then r exists and the order of a is finite

Consider the function $f(k)$
Since $a^r = 1 \bmod M$ we can write

$$f(k) = a^k \bmod M = a^{k+r} \bmod M$$

Classical period-finding

In 1994 Shor developed an algorithm for factoring integers which coupled with the quantum Fourier transform threatened to crack the standard encryption algorithms of the time

The factoring algorithm relies finding the period of a function $f(k)$

The order of an integer $a \bmod M$ is the smallest integer r such that $r > 0$ and $a^r = 1 \bmod M$

If the two integers a and M are relatively prime (i.e. they share no prime factors) then r exists and the order of a is finite

Consider the function $f(k)$
Since $a^r = 1 \bmod M$ we can write and r is the period of $f(k)$

$$f(k) = a^k \bmod M = a^{k+r} \bmod M$$

Classical period-finding

In 1994 Shor developed an algorithm for factoring integers which coupled with the quantum Fourier transform threatened to crack the standard encryption algorithms of the time

The factoring algorithm relies finding the period of a function $f(k)$

The order of an integer $a \bmod M$ is the smallest integer r such that $r > 0$ and $a^r = 1 \bmod M$

If the two integers a and M are relatively prime (i.e. they share no prime factors) then r exists and the order of a is finite

Consider the function $f(k)$
Since $a^r = 1 \bmod M$ we can write and r is the period of $f(k)$

$$f(k) = a^k \bmod M = a^{k+r} \bmod M$$

For example, take $a = 5$ and $M = 13$

Classical period-finding

In 1994 Shor developed an algorithm for factoring integers which coupled with the quantum Fourier transform threatened to crack the standard encryption algorithms of the time

The factoring algorithm relies finding the period of a function $f(k)$

The order of an integer $a \bmod M$ is the smallest integer r such that $r > 0$ and $a^r = 1 \bmod M$

If the two integers a and M are relatively prime (i.e. they share no prime factors) then r exists and the order of a is finite

Consider the function $f(k)$
Since $a^r = 1 \bmod M$ we can write and r is the period of $f(k)$

$$f(k) = a^k \bmod M = a^{k+r} \bmod M$$

r	a^r	$a^r \bmod M$

For example, take $a = 5$ and $M = 13$

Classical period-finding

In 1994 Shor developed an algorithm for factoring integers which coupled with the quantum Fourier transform threatened to crack the standard encryption algorithms of the time

The factoring algorithm relies finding the period of a function $f(k)$

The order of an integer $a \bmod M$ is the smallest integer r such that $r > 0$ and $a^r = 1 \bmod M$

If the two integers a and M are relatively prime (i.e. they share no prime factors) then r exists and the order of a is finite

Consider the function $f(k)$
Since $a^r = 1 \bmod M$ we can write and r is the period of $f(k)$

$$f(k) = a^k \bmod M = a^{k+r} \bmod M$$

For example, take $a = 5$ and $M = 13$

r	a^r	$a^r \bmod M$
1	5	5

Classical period-finding

In 1994 Shor developed an algorithm for factoring integers which coupled with the quantum Fourier transform threatened to crack the standard encryption algorithms of the time

The factoring algorithm relies finding the period of a function $f(k)$

The order of an integer $a \bmod M$ is the smallest integer r such that $r > 0$ and $a^r = 1 \bmod M$

If the two integers a and M are relatively prime (i.e. they share no prime factors) then r exists and the order of a is finite

Consider the function $f(k)$
Since $a^r = 1 \bmod M$ we can write and r is the period of $f(k)$

$$f(k) = a^k \bmod M = a^{k+r} \bmod M$$

For example, take $a = 5$ and $M = 13$

r	a^r	$a^r \bmod M$
1	5	5
2	25	12

Classical period-finding

In 1994 Shor developed an algorithm for factoring integers which coupled with the quantum Fourier transform threatened to crack the standard encryption algorithms of the time

The factoring algorithm relies finding the period of a function $f(k)$

The order of an integer $a \bmod M$ is the smallest integer r such that $r > 0$ and $a^r = 1 \bmod M$

If the two integers a and M are relatively prime (i.e. they share no prime factors) then r exists and the order of a is finite

Consider the function $f(k)$

Since $a^r = 1 \bmod M$ we can write and r is the period of $f(k)$

$$f(k) = a^k \bmod M = a^{k+r} \bmod M$$

For example, take $a = 5$ and $M = 13$

r	a^r	$a^r \bmod M$
1	5	5
2	25	12
3	125	8

Classical period-finding

In 1994 Shor developed an algorithm for factoring integers which coupled with the quantum Fourier transform threatened to crack the standard encryption algorithms of the time

The factoring algorithm relies finding the period of a function $f(k)$

The order of an integer $a \bmod M$ is the smallest integer r such that $r > 0$ and $a^r = 1 \bmod M$

If the two integers a and M are relatively prime (i.e. they share no prime factors) then r exists and the order of a is finite

Consider the function $f(k)$

Since $a^r = 1 \bmod M$ we can write and r is the period of $f(k)$

$$f(k) = a^k \bmod M = a^{k+r} \bmod M$$

For example, take $a = 5$ and $M = 13$

r	a^r	$a^r \bmod M$
1	5	5
2	25	12
3	125	8
4	625	1

Classical period-finding

In 1994 Shor developed an algorithm for factoring integers which coupled with the quantum Fourier transform threatened to crack the standard encryption algorithms of the time

The factoring algorithm relies finding the period of a function $f(k)$

The order of an integer $a \bmod M$ is the smallest integer r such that $r > 0$ and $a^r = 1 \bmod M$

If the two integers a and M are relatively prime (i.e. they share no prime factors) then r exists and the order of a is finite

Consider the function $f(k)$
Since $a^r = 1 \bmod M$ we can write

$$f(k) = a^k \bmod M = a^{k+r} \bmod M$$

and r is the period of $f(k)$

For example, take $a = 5$ and $M = 13$

r	a^r	$a^r \bmod M$
1	5	5
2	25	12
3	125	8
4	625	1

Thus $r = 4$ is the period of the function $f(k) = a^k = 5^k$

Factoring strategy

If $a^r = 1 \mod M$ and r is even then

Factoring strategy

If $a^r = 1 \mod M$ and r is even then

$$(a^{r/2} + 1)(a^{r/2} - 1) \equiv 0 \mod M$$

Factoring strategy

If $a^r = 1 \mod M$ and r is even then

$$(a^{r/2} + 1)(a^{r/2} - 1) \equiv 0 \mod M$$

In our example $r = 4$ so we have

Factoring strategy

If $a^r = 1 \mod M$ and r is even then $$(a^{r/2}+1)(a^{r/2}-1) \equiv 0 \mod M$$

In our example $r = 4$ so we have $$(5^2+1)(5^2-1)$$

Factoring strategy

If $a^r = 1 \mod M$ and r is even then

$$(a^{r/2}+1)(a^{r/2}-1) \equiv 0 \mod M$$

In our example $r = 4$ so we have

$$(5^2+1)(5^2-1) = 26 \cdot 24$$

Factoring strategy

If $a^r = 1 \mod M$ and r is even then

$$(a^{r/2}+1)(a^{r/2}-1) \equiv 0 \mod M$$

In our example $r = 4$ so we have

$$(5^2+1)(5^2-1) = 26 \cdot 24 = 13 \cdot 48$$

Factoring strategy

If $a^r = 1 \bmod M$ and r is even then $$(a^{r/2}+1)(a^{r/2}-1) \equiv 0 \mod M$$

In our example $r = 4$ so we have $$(5^2+1)(5^2-1) = 26 \cdot 24 = 13 \cdot 48$$

If neither $a^{r/2} \pm 1$ is a multiple of M then they both likely have common factors with M and so suggest a method for factoring M

Factoring strategy

If $a^r = 1 \mod M$ and r is even then $$(a^{r/2}+1)(a^{r/2}-1) \equiv 0 \mod M$$

In our example $r = 4$ so we have $$(5^2+1)(5^2-1) = 26 \cdot 24 = 13 \cdot 48$$

If neither $a^{r/2} \pm 1$ is a multiple of M then they both likely have common factors with M and so suggest a method for factoring M

1. Randomly choose an integer a and determine the period r of $f(k) = a^k \mod M$

Factoring strategy

If $a^r = 1 \mod M$ and r is even then $(a^{r/2}+1)(a^{r/2}-1) \equiv 0 \mod M$

In our example $r = 4$ so we have $(5^2+1)(5^2-1) = 26 \cdot 24 = 13 \cdot 48$

If neither $a^{r/2} \pm 1$ is a multiple of M then they both likely have common factors with M and so suggest a method for factoring M

1. Randomly choose an integer a and determine the period r of $f(k) = a^k \mod M$
2. If r is even use the Euclidean algorithm to compute the greatest common divisor of $a^{r/2} \pm 1$ and M

Factoring strategy

If $a^r = 1 \mod M$ and r is even then

$$(a^{r/2} + 1)(a^{r/2} - 1) \equiv 0 \mod M$$

In our example $r = 4$ so we have

$$(5^2 + 1)(5^2 - 1) = 26 \cdot 24 = 13 \cdot 48$$

If neither $a^{r/2} \pm 1$ is a multiple of M then they both likely have common factors with M and so suggest a method for factoring M

1. Randomly choose an integer a and determine the period r of $f(k) = a^k \mod M$
2. If r is even use the Euclidean algorithm to compute the greatest common divisor of $a^{r/2} \pm 1$ and M
3. Repeat as necessary

Factoring strategy

If $a^r = 1 \mod M$ and r is even then $(a^{r/2}+1)(a^{r/2}-1) \equiv 0 \mod M$

In our example $r=4$ so we have $(5^2+1)(5^2-1) = 26 \cdot 24 = 13 \cdot 48$

If neither $a^{r/2} \pm 1$ is a multiple of M then they both likely have common factors with M and so suggest a method for factoring M

1. Randomly choose an integer a and determine the period r of $f(k) = a^k \mod M$
2. If r is even use the Euclidean algorithm to compute the greatest common divisor of $a^{r/2} \pm 1$ and M
3. Repeat as necessary

Given that an encryption key, M, is generally a large number, this is still a computationally expensive operation for a classical computer, however Shor's quantum algorithm makes it possible efficiently perform step 2.

Shor's factoring algorithm

The implementation of Shor's algorithm can be summarized in a few steps

Shor's factoring algorithm

The implementation of Shor's algorithm can be summarized in a few steps

1. Randomly choose an integer a such that $0 < a < M$ and apply the Euclidean algorithm

Shor's factoring algorithm

The implementation of Shor's algorithm can be summarized in a few steps

1. Randomly choose an integer a such that $0 < a < M$ and apply the Euclidean algorithm
   a. If a and M have a common factor, this is a factor of M, save and start over at step 1

Shor's factoring algorithm

The implementation of Shor's algorithm can be summarized in a few steps

1. Randomly choose an integer a such that $0 < a < M$ and apply the Euclidean algorithm
   a. If a and M have a common factor, this is a factor of M, save and start over at step 1
   b. If a and M are relatively prime, continue to 2.

Shor's factoring algorithm

The implementation of Shor's algorithm can be summarized in a few steps

1. Randomly choose an integer a such that $0 < a < M$ and apply the Euclidean algorithm
   a. If a and M have a common factor, this is a factor of M, save and start over at step 1
   b. If a and M are relatively prime, continue to 2.
2. Use quantum parallelism to compute $f(x) = a^x \bmod M$ on the superposition of $n : M^2 \le 2^n < 2M^2$ inputs and apply a quantum Fourier transform to the result

Shor's factoring algorithm

The implementation of Shor's algorithm can be summarized in a few steps

1. Randomly choose an integer a such that $0 < a < M$ and apply the Euclidean algorithm
   a. If a and M have a common factor, this is a factor of M, save and start over at step 1
   b. If a and M are relatively prime, continue to 2.
2. Use quantum parallelism to compute $f(x) = a^x \bmod M$ on the superposition of $n : M^2 \leq 2^n < 2M^2$ inputs and apply a quantum Fourier transform to the result
3. Measure. With high probability, a value v close to a multiple of $\frac{2^n}{r}$ will be obtained

Shor's factoring algorithm

The implementation of Shor's algorithm can be summarized in a few steps

1. Randomly choose an integer a such that $0 < a < M$ and apply the Euclidean algorithm
   a. If a and M have a common factor, this is a factor of M, save and start over at step 1
   b. If a and M are relatively prime, continue to 2.
2. Use quantum parallelism to compute $f(x) = a^x \bmod M$ on the superposition of $n : M^2 \leq 2^n < 2M^2$ inputs and apply a quantum Fourier transform to the result
3. Measure. With high probability, a value v close to a multiple of $\frac{2^n}{r}$ will be obtained
4. Use classical methods to obtain a possible period q from v

Shor's factoring algorithm

The implementation of Shor's algorithm can be summarized in a few steps

1. Randomly choose an integer a such that $0 < a < M$ and apply the Euclidean algorithm
   a. If a and M have a common factor, this is a factor of M, save and start over at step 1
   b. If a and M are relatively prime, continue to 2.
2. Use quantum parallelism to compute $f(x) = a^x \bmod M$ on the superposition of $n : M^2 \leq 2^n < 2M^2$ inputs and apply a quantum Fourier transform to the result
3. Measure. With high probability, a value v close to a multiple of $\frac{2^n}{r}$ will be obtained
4. Use classical methods to obtain a possible period q from v
5. For q even, use the Euclidean algorithm to find any common factors of M and $a^{q/2} \pm 1$

Shor's factoring algorithm

The implementation of Shor's algorithm can be summarized in a few steps

1. Randomly choose an integer a such that $0 < a < M$ and apply the Euclidean algorithm
   a. If a and M have a common factor, this is a factor of M, save and start over at step 1
   b. If a and M are relatively prime, continue to 2.
2. Use quantum parallelism to compute $f(x) = a^x \bmod M$ on the superposition of $n : M^2 \leq 2^n < 2M^2$ inputs and apply a quantum Fourier transform to the result
3. Measure. With high probability, a value v close to a multiple of $\frac{2^n}{r}$ will be obtained
4. Use classical methods to obtain a possible period q from v
5. For q even, use the Euclidean algorithm to find any common factors of M and $a^{q/2} \pm 1$
6. Start over with step 1 if more factors are needed

Shor's factoring algorithm

The implementation of Shor's algorithm can be summarized in a few steps

1. Randomly choose an integer a such that $0 < a < M$ and apply the Euclidean algorithm
    a. If a and M have a common factor, this is a factor of M, save and start over at step 1
    b. If a and M are relatively prime, continue to 2.
2. Use quantum parallelism to compute $f(x) = a^x \bmod M$ on the superposition of $n : M^2 \le 2^n < 2M^2$ inputs and apply a quantum Fourier transform to the result
3. Measure. With high probability, a value v close to a multiple of $\frac{2^n}{r}$ will be obtained
4. Use classical methods to obtain a possible period q from v
5. For q even, use the Euclidean algorithm to find any common factors of M and $a^{q/2} \pm 1$
6. Start over with step 1 if more factors are needed

Only steps 2 and 3 require a quantum computer since the other steps are efficiently performed with a classical computer

The quantum core

Start by preparing a uniform superposition state of an n-qubit register

The quantum core

Start by preparing a uniform superposition state of an n-qubit register

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

The quantum core

Start by preparing a uniform superposition state of an n-qubit register

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

The function $f(x) = a^x \bmod M$ can be computed with an efficiently implemented transformation U_f

The quantum core

Start by preparing a uniform superposition state of an n-qubit register

$$W|0\cdots0\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle$$

The function $f(x) = a^x \bmod M$ can be computed with an efficiently implemented transformation U_f

$$U_f : |x\rangle|0\rangle \rightarrow |x\rangle|f(x)\rangle$$

The quantum core

Start by preparing a uniform superposition state of an n-qubit register

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

The function $f(x) = a^x \bmod M$ can be computed with an efficiently implemented transformation U_f

$$U_f : |x\rangle|0\rangle \rightarrow |x\rangle|f(x)\rangle$$

This requires a second m-qubit register such that

The quantum core

Start by preparing a uniform superposition state of an n-qubit register

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

The function $f(x) = a^x \bmod M$ can be computed with an efficiently implemented transformation U_f

$$U_f : |x\rangle|0\rangle \rightarrow |x\rangle|f(x)\rangle$$

This requires a second m-qubit register such that

$$U_f \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|0\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|f(x)\rangle$$

The quantum core

Start by preparing a uniform superposition state of an n-qubit register

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

The function $f(x) = a^x \bmod M$ can be computed with an efficiently implemented transformation U_f

$$U_f : |x\rangle|0\rangle \to |x\rangle|f(x)\rangle$$

This requires a second m-qubit register such that

$$U_f \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|0\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|f(x)\rangle$$

The second register is now measured randomly and this returns a value u for $f(x)$ so that the two registers are no longer entangled and the state is

The quantum core

Start by preparing a uniform superposition state of an n-qubit register

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

The function $f(x) = a^x \bmod M$ can be computed with an efficiently implemented transformation U_f

$$U_f : |x\rangle|0\rangle \to |x\rangle|f(x)\rangle$$

This requires a second m-qubit register such that

$$U_f \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|0\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|f(x)\rangle$$

The second register is now measured randomly and this returns a value u for $f(x)$ so that the two registers are no longer entangled and the state is

$$C\sum_{x=0}^{N-1} g(x)|x\rangle|u\rangle, \quad g(x) = \begin{cases} 1 & \text{if } f(x) = u \\ 0 & \text{otherwise} \end{cases}$$

The quantum core

Start by preparing a uniform superposition state of an n-qubit register

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

The function $f(x) = a^x \bmod M$ can be computed with an efficiently implemented transformation U_f

$$U_f : |x\rangle|0\rangle \to |x\rangle|f(x)\rangle$$

This requires a second m-qubit register such that

$$U_f \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|0\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|f(x)\rangle$$

The second register is now measured randomly and this returns a value u for $f(x)$ so that the two registers are no longer entangled and the state is

$$C\sum_{x=0}^{N-1}g(x)|x\rangle|u\rangle, \quad g(x) = \begin{cases} 1 & \text{if } f(x) = u \\ 0 & \text{otherwise} \end{cases}$$

C is the normalization constant and $g(x)$ must, by definition, have the same period as $f(x)$ but is sparse and only has non-zero values at intervals of the period

Applying the quantum Fourier transform

The second register can be thrown away as it is no longer entangled with the first

Applying the quantum Fourier transform

The second register can be thrown away as it is no longer entangled with the first

$$|\psi\rangle = C \sum_{x=0}^{N-1} g(x)|x\rangle$$

Applying the quantum Fourier transform

The second register can be thrown away as it is no longer entangled with the first

$$|\psi\rangle = C \sum_{x=0}^{N-1} g(x)|x\rangle$$

Now apply the quantum Fourier transform, U_F, to the first register to get

Applying the quantum Fourier transform

The second register can be thrown away as it is no longer entangled with the first

$$|\psi\rangle = C \sum_{x=0}^{N-1} g(x)|x\rangle$$

Now apply the quantum Fourier transform, U_F, to the first register to get

$$U_F|\psi\rangle = C' \sum_{c=0}^{N-1} G(c)|c\rangle$$

Applying the quantum Fourier transform

The second register can be thrown away as it is no longer entangled with the first

$$|\psi\rangle = C \sum_{x=0}^{N-1} g(x)|x\rangle$$

Now apply the quantum Fourier transform, U_F, to the first register to get

$$U_F|\psi\rangle = C' \sum_{c=0}^{N-1} G(c)|c\rangle$$

Where $G(c)$ is given by

Applying the quantum Fourier transform

The second register can be thrown away as it is no longer entangled with the first

$$|\psi\rangle = C \sum_{x=0}^{N-1} g(x)|x\rangle$$

Now apply the quantum Fourier transform, U_F, to the first register to get

$$U_F|\psi\rangle = C' \sum_{c=0}^{N-1} G(c)|c\rangle$$

Where $G(c)$ is given by

$$G(c) = \sum_{x=0}^{N-1} g(x)e^{2\pi icx/2^n}$$

Applying the quantum Fourier transform

The second register can be thrown away as it is no longer entangled with the first

$$|\psi\rangle = C \sum_{x=0}^{N-1} g(x)|x\rangle$$

Now apply the quantum Fourier transform, U_F, to the first register to get

$$U_F|\psi\rangle = C' \sum_{c=0}^{N-1} G(c)|c\rangle$$

Where $G(c)$ is given by

$$G(c) = \sum_{x=0}^{N-1} g(x) e^{2\pi i c x/2^n}$$

Recalling the properties of the quantum Fourier transform, if the period, r, of the function $g(x)$ is a power of two, $G(c) \equiv 0$ except when c is a multiple of $\frac{2^n}{r}$

Applying the quantum Fourier transform

The second register can be thrown away as it is no longer entangled with the first

$$|\psi\rangle = C \sum_{x=0}^{N-1} g(x)|x\rangle$$

Now apply the quantum Fourier transform, U_F, to the first register to get

$$U_F|\psi\rangle = C' \sum_{c=0}^{N-1} G(c)|c\rangle$$

Where $G(c)$ is given by

$$G(c) = \sum_{x=0}^{N-1} g(x)e^{2\pi icx/2^n}$$

Recalling the properties of the quantum Fourier transform, if the period, r, of the function $g(x)$ is a power of two, $G(c) \equiv 0$ except when c is a multiple of $\frac{2^n}{r}$

When the period is not a power of two, the quantum Fourier transform approximates the exact case and yields a value v close to a multiple of $\frac{2^n}{r}$

Continued fraction expansion

In the case where r is a power of 2,the measured output $v = j\frac{2^n}{r}$ and the period is straightforward to extract

Continued fraction expansion

In the case where r is a power of 2,the measured output $v = j\frac{2^n}{r}$ and the period is straightforward to extract

It is more challenging when the Fourier transform produces values which are only approximate multiples of the scaled frequency

Continued fraction expansion

In the case where r is a power of 2,the measured output $v = j\frac{2^n}{r}$ and the period is straightforward to extract

It is more challenging when the Fourier transform produces values which are only approximate multiples of the scaled frequency

In this case, a good guess for the period is obtained by the continued fraction expansion of $\frac{v}{2^n}$

Continued fraction expansion

In the case where r is a power of 2,the measured output $v = j\frac{2^n}{r}$ and the period is straightforward to extract

It is more challenging when the Fourier transform produces values which are only approximate multiples of the scaled frequency

In this case, a good guess for the period is obtained by the continued fraction expansion of $\frac{v}{2^n}$

Define $[x] = trunc(x)$ as the greatest integer less than x and define the quantities

Continued fraction expansion

In the case where r is a power of 2,the measured output $v = j\frac{2^n}{r}$ and the period is straightforward to extract

It is more challenging when the Fourier transform produces values which are only approximate multiples of the scaled frequency

In this case, a good guess for the period is obtained by the continued fraction expansion of $\frac{v}{2^n}$

Define $[x] = trunc(x)$ as the greatest integer less than x and define the quantities

$$a_0 = \left[\frac{v}{2^n}\right],$$

Continued fraction expansion

In the case where r is a power of 2,the measured output $v = j\frac{2^n}{r}$ and the period is straightforward to extract

It is more challenging when the Fourier transform produces values which are only approximate multiples of the scaled frequency

In this case, a good guess for the period is obtained by the continued fraction expansion of $\frac{v}{2^n}$

Define $[x] = trunc(x)$ as the greatest integer less than x and define the quantities

$$a_0 = \left[\frac{v}{2^n}\right], \qquad \epsilon_0 = \frac{v}{2^n} - a_0,$$

Continued fraction expansion

In the case where r is a power of 2,the measured output $v = j\frac{2^n}{r}$ and the period is straightforward to extract

It is more challenging when the Fourier transform produces values which are only approximate multiples of the scaled frequency

In this case, a good guess for the period is obtained by the continued fraction expansion of $\frac{v}{2^n}$

Define $[x] = trunc(x)$ as the greatest integer less than x and define the quantities

$$a_0 = \left[\frac{v}{2^n}\right], \quad \epsilon_0 = \frac{v}{2^n} - a_0, \quad a_i = \left[\frac{1}{\epsilon_{i-1}}\right],$$

Continued fraction expansion

In the case where r is a power of 2,the measured output $v = j\frac{2^n}{r}$ and the period is straightforward to extract

It is more challenging when the Fourier transform produces values which are only approximate multiples of the scaled frequency

In this case, a good guess for the period is obtained by the continued fraction expansion of $\frac{v}{2^n}$

Define $[x] = trunc(x)$ as the greatest integer less than x and define the quantities

$$a_0 = \left[\frac{v}{2^n}\right], \quad \epsilon_0 = \frac{v}{2^n} - a_0, \quad a_i = \left[\frac{1}{\epsilon_{i-1}}\right], \quad \epsilon_i = \frac{1}{\epsilon_{i-1}} - a_i$$

Continued fraction expansion

In the case where r is a power of 2,the measured output $v = j\frac{2^n}{r}$ and the period is straightforward to extract

It is more challenging when the Fourier transform produces values which are only approximate multiples of the scaled frequency

In this case, a good guess for the period is obtained by the continued fraction expansion of $\frac{v}{2^n}$

Define $[x] = trunc(x)$ as the greatest integer less than x and define the quantities

$$a_0 = \left[\frac{v}{2^n}\right], \quad \epsilon_0 = \frac{v}{2^n} - a_0, \quad a_i = \left[\frac{1}{\epsilon_{i-1}}\right], \quad \epsilon_i = \frac{1}{\epsilon_{i-1}} - a_i$$

$p_0 = a_0,$

Continued fraction expansion

In the case where r is a power of 2,the measured output $v = j\frac{2^n}{r}$ and the period is straightforward to extract

It is more challenging when the Fourier transform produces values which are only approximate multiples of the scaled frequency

In this case, a good guess for the period is obtained by the continued fraction expansion of $\frac{v}{2^n}$

Define $[x] = trunc(x)$ as the greatest integer less than x and define the quantities

$$a_0 = \left[\frac{v}{2^n}\right], \quad \epsilon_0 = \frac{v}{2^n} - a_0, \quad a_i = \left[\frac{1}{\epsilon_{i-1}}\right], \quad \epsilon_i = \frac{1}{\epsilon_{i-1}} - a_i$$

$p_0 = a_0, \quad p_1 = a_1 a_0 + 1,$

Continued fraction expansion

In the case where r is a power of 2,the measured output $v = j\frac{2^n}{r}$ and the period is straightforward to extract

It is more challenging when the Fourier transform produces values which are only approximate multiples of the scaled frequency

In this case, a good guess for the period is obtained by the continued fraction expansion of $\frac{v}{2^n}$

Define $[x] = trunc(x)$ as the greatest integer less than x and define the quantities

$$a_0 = \left[\frac{v}{2^n}\right], \qquad \epsilon_0 = \frac{v}{2^n} - a_0, \qquad a_i = \left[\frac{1}{\epsilon_{i-1}}\right], \qquad \epsilon_i = \frac{1}{\epsilon_{i-1}} - a_i$$

$$p_0 = a_0, \qquad p_1 = a_1 a_0 + 1, \qquad p_i = a_i p_{i-1} + p_{i-2},$$

Continued fraction expansion

In the case where r is a power of 2,the measured output $v = j\frac{2^n}{r}$ and the period is straightforward to extract

It is more challenging when the Fourier transform produces values which are only approximate multiples of the scaled frequency

In this case, a good guess for the period is obtained by the continued fraction expansion of $\frac{v}{2^n}$

Define $[x] = trunc(x)$ as the greatest integer less than x and define the quantities

$$a_0 = \left[\frac{v}{2^n}\right], \quad \epsilon_0 = \frac{v}{2^n} - a_0, \quad a_i = \left[\frac{1}{\epsilon_{i-1}}\right], \quad \epsilon_i = \frac{1}{\epsilon_{i-1}} - a_i$$

$$p_0 = a_0, \quad p_1 = a_1 a_0 + 1, \quad p_i = a_i p_{i-1} + p_{i-2}, \quad q_0 = 1,$$

Continued fraction expansion

In the case where r is a power of 2,the measured output $v = j\frac{2^n}{r}$ and the period is straightforward to extract

It is more challenging when the Fourier transform produces values which are only approximate multiples of the scaled frequency

In this case, a good guess for the period is obtained by the continued fraction expansion of $\frac{v}{2^n}$

Define $[x] = trunc(x)$ as the greatest integer less than x and define the quantities

$$a_0 = \left[\frac{v}{2^n}\right], \quad \epsilon_0 = \frac{v}{2^n} - a_0, \quad a_i = \left[\frac{1}{\epsilon_{i-1}}\right], \quad \epsilon_i = \frac{1}{\epsilon_{i-1}} - a_i$$

$$p_0 = a_0, \quad p_1 = a_1 a_0 + 1, \quad p_i = a_i p_{i-1} + p_{i-2}, \quad q_0 = 1, \quad q_1 = a_1,$$

Continued fraction expansion

In the case where r is a power of 2,the measured output $v = j\frac{2^n}{r}$ and the period is straightforward to extract

It is more challenging when the Fourier transform produces values which are only approximate multiples of the scaled frequency

In this case, a good guess for the period is obtained by the continued fraction expansion of $\frac{v}{2^n}$

Define $[x] = trunc(x)$ as the greatest integer less than x and define the quantities

$$a_0 = \left[\frac{v}{2^n}\right], \quad \epsilon_0 = \frac{v}{2^n} - a_0, \quad a_i = \left[\frac{1}{\epsilon_{i-1}}\right], \quad \epsilon_i = \frac{1}{\epsilon_{i-1}} - a_i$$

$$p_0 = a_0, \quad p_1 = a_1 a_0 + 1, \quad p_i = a_i p_{i-1} + p_{i-2}, \quad q_0 = 1, \quad q_1 = a_1, \quad q_i = a_i q_{i-1} + q_{i-2}$$

Continued fraction expansion

In the case where r is a power of 2,the measured output $v = j\frac{2^n}{r}$ and the period is straightforward to extract

It is more challenging when the Fourier transform produces values which are only approximate multiples of the scaled frequency

In this case, a good guess for the period is obtained by the continued fraction expansion of $\frac{v}{2^n}$

Define $[x] = trunc(x)$ as the greatest integer less than x and define the quantities

$$a_0 = \left[\frac{v}{2^n}\right], \quad \epsilon_0 = \frac{v}{2^n} - a_0, \quad a_i = \left[\frac{1}{\epsilon_{i-1}}\right], \quad \epsilon_i = \frac{1}{\epsilon_{i-1}} - a_i$$

$$p_0 = a_0, \quad p_1 = a_1 a_0 + 1, \quad p_i = a_i p_{i-1} + p_{i-2}, \quad q_0 = 1, \quad q_1 = a_1, \quad q_i = a_i q_{i-1} + q_{i-2}$$

Compute the first fraction $\frac{p_i}{q_i}$ such that $q_i < M \leq q_{i+1}$

Continued fraction expansion

In the case where r is a power of 2,the measured output $v = j\frac{2^n}{r}$ and the period is straightforward to extract

It is more challenging when the Fourier transform produces values which are only approximate multiples of the scaled frequency

In this case, a good guess for the period is obtained by the continued fraction expansion of $\frac{v}{2^n}$

Define $[x] = trunc(x)$ as the greatest integer less than x and define the quantities

$$a_0 = \left[\frac{v}{2^n}\right], \quad \epsilon_0 = \frac{v}{2^n} - a_0, \quad a_i = \left[\frac{1}{\epsilon_{i-1}}\right], \quad \epsilon_i = \frac{1}{\epsilon_{i-1}} - a_i$$

$$p_0 = a_0, \quad p_1 = a_1 a_0 + 1, \quad p_i = a_i p_{i-1} + p_{i-2}, \quad q_0 = 1, \quad q_1 = a_1, \quad q_i = a_i q_{i-1} + q_{i-2}$$

Compute the first fraction $\frac{p_i}{q_i}$ such that $q_i < M \leq q_{i+1}$

This is the unique fraction with denominator less than M that is within $\frac{1}{M^2}$ of $\frac{v}{2^n}$

Continued fraction expansion

In the case where r is a power of 2,the measured output $v = j\frac{2^n}{r}$ and the period is straightforward to extract

It is more challenging when the Fourier transform produces values which are only approximate multiples of the scaled frequency

In this case, a good guess for the period is obtained by the continued fraction expansion of $\frac{v}{2^n}$

Define $[x] = trunc(x)$ as the greatest integer less than x and define the quantities

$$a_0 = \left[\frac{v}{2^n}\right], \quad \epsilon_0 = \frac{v}{2^n} - a_0, \quad a_i = \left[\frac{1}{\epsilon_{i-1}}\right], \quad \epsilon_i = \frac{1}{\epsilon_{i-1}} - a_i$$

$$p_0 = a_0, \quad p_1 = a_1 a_0 + 1, \quad p_i = a_i p_{i-1} + p_{i-2}, \quad q_0 = 1, \quad q_1 = a_1, \quad q_i = a_i q_{i-1} + q_{i-2}$$

Compute the first fraction $\frac{p_i}{q_i}$ such that $q_i < M \leq q_{i+1}$

This is the unique fraction with denominator less than M that is within $\frac{1}{M^2}$ of $\frac{v}{2^n}$

Shor showed that this fraction is within $\frac{1}{2}$ of a multiple of $\frac{2^n}{r}$

Period extraction

Recall that we chose the size of the qubit register to be $n : M^2 \leq 2^n < 2M^2$

Period extraction

Recall that we chose the size of the qubit register to be $n : M^2 \leq 2^n < 2M^2$

According to Shor, in the high probability case that

Period extraction

Recall that we chose the size of the qubit register to be $n : M^2 \leq 2^n < 2M^2$

According to Shor, in the high probability case that

$$\left| v - j\frac{2^n}{r} \right| < \frac{1}{2}$$

Period extraction

Recall that we chose the size of the qubit register to be $n : M^2 \leq 2^n < 2M^2$

According to Shor, in the high probability case that

$$\left| v - j\frac{2^n}{r} \right| < \frac{1}{2}$$

For some j, $M^2 \leq 2^n$ so that

Period extraction

Recall that we chose the size of the qubit register to be $n : M^2 \leq 2^n < 2M^2$

According to Shor, in the high probability case that

$$\left| v - j\frac{2^n}{r} \right| < \frac{1}{2}$$

For some j, $M^2 \leq 2^n$ so that

$$\left| \frac{v}{2^n} - \frac{j}{r} \right| < \frac{1}{2 \cdot 2^n}$$

Period extraction

Recall that we chose the size of the qubit register to be $n : M^2 \leq 2^n < 2M^2$

According to Shor, in the high probability case that

$$\left| v - j\frac{2^n}{r} \right| < \frac{1}{2}$$

For some j, $M^2 \leq 2^n$ so that

$$\left| \frac{v}{2^n} - \frac{j}{r} \right| < \frac{1}{2 \cdot 2^n} \leq \frac{1}{2M^2}$$

Period extraction

Recall that we chose the size of the qubit register to be $n : M^2 \leq 2^n < 2M^2$

According to Shor, in the high probability case that

$$\left| v - j\frac{2^n}{r} \right| < \frac{1}{2}$$

For some j, $M^2 \leq 2^n$ so that

$$\left| \frac{v}{2^n} - \frac{j}{r} \right| < \frac{1}{2 \cdot 2^n} \leq \frac{1}{2M^2}$$

The difference between two fractions $\frac{p}{q}$ and $\frac{p'}{q'}$ with denominators less than M is bounded

Period extraction

Recall that we chose the size of the qubit register to be $n : M^2 \leq 2^n < 2M^2$

According to Shor, in the high probability case that

$$\left| v - j\frac{2^n}{r} \right| < \frac{1}{2}$$

For some j, $M^2 \leq 2^n$ so that

$$\left| \frac{v}{2^n} - \frac{j}{r} \right| < \frac{1}{2 \cdot 2^n} \leq \frac{1}{2M^2}$$

The difference between two fractions $\frac{p}{q}$ and $\frac{p'}{q'}$ with denominators less than M is bounded

$$\left| \frac{p}{q} - \frac{p'}{q'} \right| = \left| \frac{pq' - p'q}{qq'} \right| > \frac{1}{M^2}$$

Period extraction

Recall that we chose the size of the qubit register to be $n : M^2 \leq 2^n < 2M^2$

According to Shor, in the high probability case that

$$\left| v - j\frac{2^n}{r} \right| < \frac{1}{2}$$

For some j, $M^2 \leq 2^n$ so that

$$\left| \frac{v}{2^n} - \frac{j}{r} \right| < \frac{1}{2 \cdot 2^n} \leq \frac{1}{2M^2}$$

The difference between two fractions $\frac{p}{q}$ and $\frac{p'}{q'}$ with denominators less than M is bounded

$$\left| \frac{p}{q} - \frac{p'}{q'} \right| = \left| \frac{pq' - p'q}{qq'} \right| > \frac{1}{M^2}$$

There is at most one fraction $\frac{p}{q}$ with denominator $q < M$ such that

Period extraction

Recall that we chose the size of the qubit register to be $n : M^2 \leq 2^n < 2M^2$

According to Shor, in the high probability case that

$$\left| v - j\frac{2^n}{r} \right| < \frac{1}{2}$$

For some j, $M^2 \leq 2^n$ so that

$$\left| \frac{v}{2^n} - \frac{j}{r} \right| < \frac{1}{2 \cdot 2^n} \leq \frac{1}{2M^2}$$

The difference between two fractions $\frac{p}{q}$ and $\frac{p'}{q'}$ with denominators less than M is bounded

$$\left| \frac{p}{q} - \frac{p'}{q'} \right| = \left| \frac{pq' - p'q}{qq'} \right| > \frac{1}{M^2}$$

There is at most one fraction $\frac{p}{q}$ with denominator $q < M$ such that

$$\left| \frac{v}{2^n} - \frac{p}{q} \right| < \frac{1}{M^2}$$

Period extraction

Recall that we chose the size of the qubit register to be $n : M^2 \leq 2^n < 2M^2$

According to Shor, in the high probability case that

$$\left| v - j\frac{2^n}{r} \right| < \frac{1}{2}$$

For some j, $M^2 \leq 2^n$ so that

$$\left| \frac{v}{2^n} - \frac{j}{r} \right| < \frac{1}{2 \cdot 2^n} \leq \frac{1}{2M^2}$$

The difference between two fractions $\frac{p}{q}$ and $\frac{p'}{q'}$ with denominators less than M is bounded

$$\left| \frac{p}{q} - \frac{p'}{q'} \right| = \left| \frac{pq' - p'q}{qq'} \right| > \frac{1}{M^2}$$

There is at most one fraction $\frac{p}{q}$ with denominator $q < M$ such that

$$\left| \frac{v}{2^n} - \frac{p}{q} \right| < \frac{1}{M^2}$$

This fraction, computed by fraction expansion will likely be equal to $\frac{j}{r}$ so the denominator q is the guess for the period r which will be correct if r and j are relatively prime

Shor's algorithm example

In order to factor $M = 21$, note that $M^2 = 441$ so that $2^9 = 512$ is the power of 2 between M^2 and M^2

Shor's algorithm example

In order to factor $M = 21$, note that $M^2 = 441$ so that $2^9 = 512$ is the power of 2 between M^2 and M^2

With $n = 9$ as the size of the first register, the size of the second is set by the ceiling $\lceil \ln M \rceil + 1 = m = 5$

Shor's algorithm example

In order to factor $M = 21$, note that $M^2 = 441$ so that $2^9 = 512$ is the power of 2 between M^2 and M^2

With $n = 9$ as the size of the first register, the size of the second is set by the ceiling $\lceil \ln M \rceil + 1 = m = 5$

The state, after applying U_f is therefore,

Shor's algorithm example

In order to factor $M = 21$, note that $M^2 = 441$ so that $2^9 = 512$ is the power of 2 between M^2 and M^2

With $n = 9$ as the size of the first register, the size of the second is set by the ceiling $\lceil \ln M \rceil + 1 = m = 5$

The state, after applying U_f is therefore,

$$|\psi\rangle = \frac{1}{\sqrt{2^9}} \sum_{x=0}^{2^9-1} |x\rangle |f(x)\rangle$$

Shor's algorithm example

In order to factor $M = 21$, note that $M^2 = 441$ so that $2^9 = 512$ is the power of 2 between M^2 and M^2

With $n = 9$ as the size of the first register, the size of the second is set by the ceiling $\lceil \ln M \rceil + 1 = m = 5$

The state, after applying U_f is therefore, a 14-qubit state with 9 qubits in the first register and 5 in the second

$$|\psi\rangle = \frac{1}{\sqrt{2^9}} \sum_{x=0}^{2^9-1} |x\rangle |f(x)\rangle$$

Shor's algorithm example

In order to factor $M = 21$, note that $M^2 = 441$ so that $2^9 = 512$ is the power of 2 between M^2 and M^2

With $n = 9$ as the size of the first register, the size of the second is set by the ceiling $\lceil \ln M \rceil + 1 = m = 5$

The state, after applying U_f is therefore, a 14-qubit state with 9 qubits in the first register and 5 in the second

$$|\psi\rangle = \frac{1}{\sqrt{2^9}} \sum_{x=0}^{2^9-1} |x\rangle|f(x)\rangle$$

If the randomly selected integer $a = 11$ and the measurement of the second register gives $u = 8$

Shor's algorithm example

In order to factor $M = 21$, note that $M^2 = 441$ so that $2^9 = 512$ is the power of 2 between M^2 and M^2

With $n = 9$ as the size of the first register, the size of the second is set by the ceiling $\lceil \ln M \rceil + 1 = m = 5$

The state, after applying U_f is therefore, a 14-qubit state with 9 qubits in the first register and 5 in the second

$$|\psi\rangle = \frac{1}{\sqrt{2^9}} \sum_{x=0}^{2^9-1} |x\rangle |f(x)\rangle$$

If the randomly selected integer $a = 11$ and the measurement of the second register gives $u = 8$

The state of the first register after the measurement shows the periodicity of $f(x)$

Shor's algorithm example

In order to factor $M = 21$, note that $M^2 = 441$ so that $2^9 = 512$ is the power of 2 between M^2 and M^2

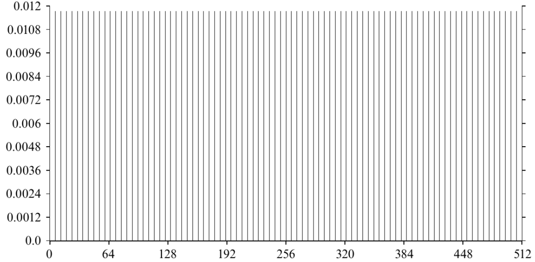
With $n = 9$ as the size of the first register, the size of the second is set by the ceiling $\lceil \ln M \rceil + 1 = m = 5$

The state, after applying U_f is therefore, a 14-qubit state with 9 qubits in the first register and 5 in the second

$$|\psi\rangle = \frac{1}{\sqrt{2^9}} \sum_{x=0}^{2^9-1} |x\rangle |f(x)\rangle$$

If the randomly selected integer $a = 11$ and the measurement of the second register gives $u = 8$

The state of the first register after the measurement shows the periodicity of $f(x)$

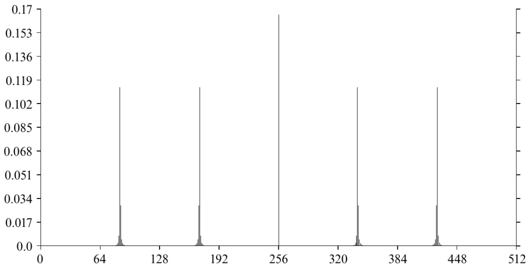


Shor's algorithm example

The result of the Fourier transform U_F is applied to $|\psi\rangle$ clearly shows that the period of $f(x)$ is not a multiple of 2

Shor's algorithm example

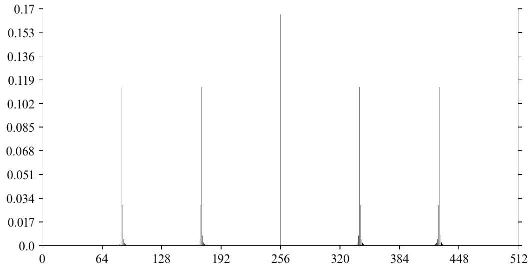
The result of the Fourier transform U_F is applied to $|\psi\rangle$ clearly shows that the period of $f(x)$ is not a multiple of 2



Shor's algorithm example

The result of the Fourier transform U_F is applied to $|\psi\rangle$ clearly shows that the period of $f(x)$ is not a multiple of 2

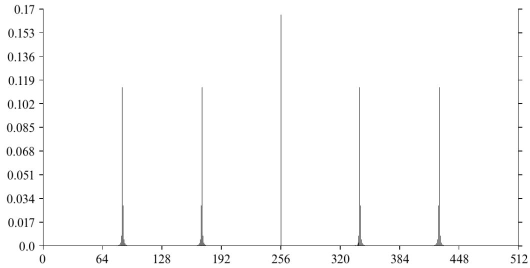
Measurement of $|\psi\rangle$ now returns a value $v = 427$ which is relative prime to 2^n



Shor's algorithm example

The result of the Fourier transform U_F is applied to $|\psi\rangle$ clearly shows that the period of $f(x)$ is not a multiple of 2

Measurement of $|\psi\rangle$ now returns a value $v = 427$ which is relative prime to 2^n

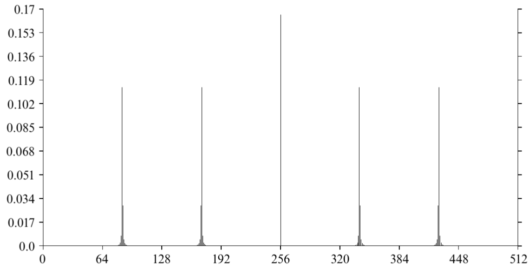


The continued fraction algorithm is then applied, giving

Shor's algorithm example

The result of the Fourier transform U_F is applied to $|\psi\rangle$ clearly shows that the period of $f(x)$ is not a multiple of 2

Measurement of $|\psi\rangle$ now returns a value $v = 427$ which is relative prime to 2^n



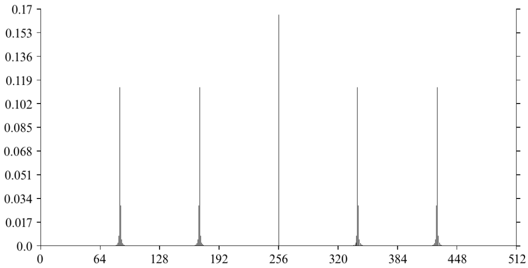
The continued fraction algorithm is then applied, giving

i	a_i	p_i	q_i	ϵ_i

Shor's algorithm example

The result of the Fourier transform U_F is applied to $|\psi\rangle$ clearly shows that the period of $f(x)$ is not a multiple of 2

Measurement of $|\psi\rangle$ now returns a value $v = 427$ which is relative prime to 2^n



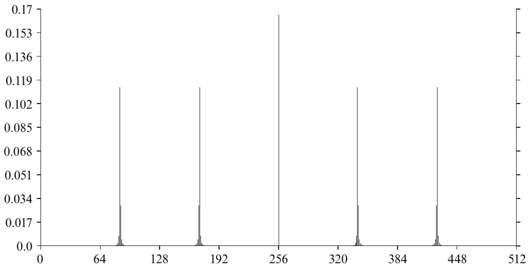
The continued fraction algorithm is then applied, giving

i	a_i	p_i	q_i	ϵ_i
0	0	0	1	0.8339844

Shor's algorithm example

The result of the Fourier transform U_F is applied to $|\psi\rangle$ clearly shows that the period of $f(x)$ is not a multiple of 2

Measurement of $|\psi\rangle$ now returns a value $v = 427$ which is relative prime to 2^n



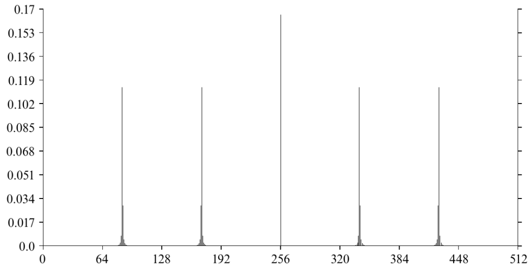
The continued fraction algorithm is then applied, giving

i	a_i	p_i	q_i	ϵ_i
0	0	0	1	0.8339844
1	1	1	1	0.1990632

Shor's algorithm example

The result of the Fourier transform U_F is applied to $|\psi\rangle$ clearly shows that the period of $f(x)$ is not a multiple of 2

Measurement of $|\psi\rangle$ now returns a value $v = 427$ which is relative prime to 2^n



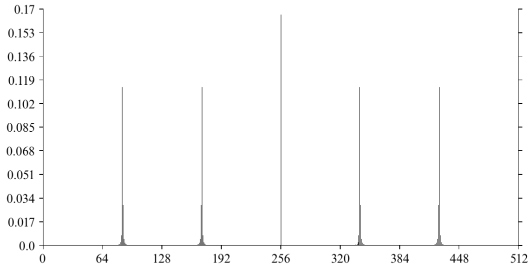
The continued fraction algorithm is then applied, giving

i	a_i	p_i	q_i	ϵ_i
0	0	0	1	0.8339844
1	1	1	1	0.1990632
2	5	5	6	0.02352941

Shor's algorithm example

The result of the Fourier transform U_F is applied to $|\psi\rangle$ clearly shows that the period of $f(x)$ is not a multiple of 2

Measurement of $|\psi\rangle$ now returns a value $v = 427$ which is relative prime to 2^n



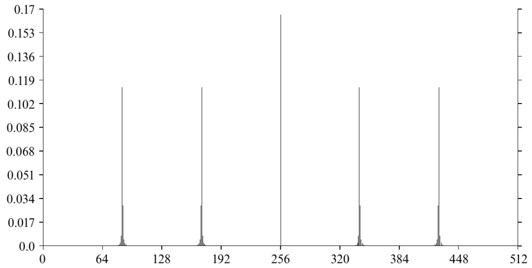
The continued fraction algorithm is then applied, giving

i	a_i	p_i	q_i	ϵ_i
0	0	0	1	0.8339844
1	1	1	1	0.1990632
2	5	5	6	0.02352941
3	42	211	253	0.5

Shor's algorithm example

The result of the Fourier transform U_F is applied to $|\psi\rangle$ clearly shows that the period of $f(x)$ is not a multiple of 2

Measurement of $|\psi\rangle$ now returns a value $v = 427$ which is relative prime to 2^n



The continued fraction algorithm is then applied, giving

i	a_i	p_i	q_i	ϵ_i
0	0	0	1	0.8339844
1	1	1	1	0.1990632
2	5	5	6	0.02352941
3	42	211	253	0.5

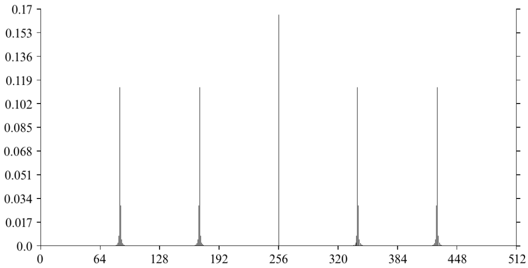
The computation is terminated when

$6 = q_2 < M \leq q_3 = 253$ since $M = 21$

Shor's algorithm example

The result of the Fourier transform U_F is applied to $|\psi\rangle$ clearly shows that the period of $f(x)$ is not a multiple of 2

Measurement of $|\psi\rangle$ now returns a value $v = 427$ which is relative prime to 2^n



The continued fraction algorithm is then applied, giving

i	a_i	p_i	q_i	ϵ_i
0	0	0	1	0.8339844
1	1	1	1	0.1990632
2	5	5	6	0.02352941
3	42	211	253	0.5

The computation is terminated when $6 = q_2 < M \leq q_3 = 253$ since $M = 21$

$q = 6$ is thus the guess for the period of $f(x)$

Shor's algorithm example

With $q = 6$ being even we can now find the greatest common factor of $a^{q/2} \pm 1$ and M where $M = 21$ and $a = 11$ by applying the Euclidean algorithm

Shor's algorithm example

With $q = 6$ being even we can now find the greatest common factor of $a^{q/2} \pm 1$ and M where $M = 21$ and $a = 11$ by applying the Euclidean algorithm

$$a^{q/2} + 1 = 11^3 + 1 = 1332$$

Shor's algorithm example

With $q = 6$ being even we can now find the greatest common factor of $a^{q/2} \pm 1$ and M where $M = 21$ and $a = 11$ by applying the Euclidean algorithm

$$a^{q/2} + 1 = 11^3 + 1 = 1332$$

M n m

Shor's algorithm example

With $q = 6$ being even we can now find the greatest common factor of $a^{q/2} \pm 1$ and M where $M = 21$ and $a = 11$ by applying the Euclidean algorithm

$$a^{q/2} + 1 = 11^3 + 1 = 1332$$

	M	n	m
1332	21		63

Shor's algorithm example

With $q = 6$ being even we can now find the greatest common factor of $a^{q/2} \pm 1$ and M where $M = 21$ and $a = 11$ by applying the Euclidean algorithm

$$a^{q/2} + 1 = 11^3 + 1 = 1332$$

	M	n	m
1332	21		63
9	21	2	

Shor's algorithm example

With $q = 6$ being even we can now find the greatest common factor of $a^{q/2} \pm 1$ and M where $M = 21$ and $a = 11$ by applying the Euclidean algorithm

$$a^{q/2} + 1 = 11^3 + 1 = 1332$$

	M	n	m
1332	21		63
9	21	2	
9	3		3

Shor's algorithm example

With $q = 6$ being even we can now find the greatest common factor of $a^{q/2} \pm 1$ and M where $M = 21$ and $a = 11$ by applying the Euclidean algorithm

$$a^{q/2} + 1 = 11^3 + 1 = 1332$$

	M	n	m
1332	21		63
9	21	2	
9	3		3
0			

Shor's algorithm example

With $q = 6$ being even we can now find the greatest common factor of $a^{q/2} \pm 1$ and M where $M = 21$ and $a = 11$ by applying the Euclidean algorithm

$$a^{q/2} + 1 = 11^3 + 1 = 1332$$

$$a^{q/2} - 1 = 11^3 + 1 = 1330$$

	M	n	m
1332	21		63
9	21	2	
9	3		3
0			

Shor's algorithm example

With $q = 6$ being even we can now find the greatest common factor of $a^{q/2} \pm 1$ and M where $M = 21$ and $a = 11$ by applying the Euclidean algorithm

$$a^{q/2} + 1 = 11^3 + 1 = 1332$$

	M	n	m
1332	21		63
9	21	2	
9	3		3
0			

$$a^{q/2} - 1 = 11^3 + 1 = 1330$$

M	n	m

Shor's algorithm example

With $q = 6$ being even we can now find the greatest common factor of $a^{q/2} \pm 1$ and M where $M = 21$ and $a = 11$ by applying the Euclidean algorithm

$$a^{q/2} + 1 = 11^3 + 1 = 1332$$

	M	n	m
1332	21		63
9	21	2	
9	3		3
0			

$$a^{q/2} - 1 = 11^3 + 1 = 1330$$

	M	n	m
1330	21		63

Shor's algorithm example

With $q = 6$ being even we can now find the greatest common factor of $a^{q/2} \pm 1$ and M where $M = 21$ and $a = 11$ by applying the Euclidean algorithm

$$a^{q/2} + 1 = 11^3 + 1 = 1332$$

	M	n	m
1332	21		63
9	21	2	
9	3		3
0			

$$a^{q/2} - 1 = 11^3 + 1 = 1330$$

	M	n	m
1330	21		63
7	21	3	

Shor's algorithm example

With $q = 6$ being even we can now find the greatest common factor of $a^{q/2} \pm 1$ and M where $M = 21$ and $a = 11$ by applying the Euclidean algorithm

$$a^{q/2} + 1 = 11^3 + 1 = 1332$$

	M	n	m
1332	21		63
9	21	2	
9	3		3
0			

$$a^{q/2} - 1 = 11^3 + 1 = 1330$$

	M	n	m
1330	21		63
7	21	3	
	0		

Shor's algorithm example

With $q = 6$ being even we can now find the greatest common factor of $a^{q/2} \pm 1$ and M where $M = 21$ and $a = 11$ by applying the Euclidean algorithm

$$a^{q/2} + 1 = 11^3 + 1 = 1332$$

	M	n	m
1332	21		63
9	21	2	
9	3		3
0			

$$a^{q/2} - 1 = 11^3 + 1 = 1330$$

	M	n	m
1330	21		63
7	21	3	
	0		

With a single Fourier transform application we have factored $M = 21$ into 3 and 7

Shor's algorithm example

With $q = 6$ being even we can now find the greatest common factor of $a^{q/2} \pm 1$ and M where $M = 21$ and $a = 11$ by applying the Euclidean algorithm

$$a^{q/2} + 1 = 11^3 + 1 = 1332$$

	M	n	m
1332	21		63
9	21	2	
9	3		3
0			

$$a^{q/2} - 1 = 11^3 + 1 = 1330$$

	M	n	m
1330	21		63
7	21	3	
	0		

With a single Fourier transform application we have factored $M = 21$ into 3 and 7

Clearly this is a trivial example but the potential efficiency of the algorithm is evident