Procesamiento Cuántico de Datos

Miguel Arizmendi, Gustavo Zabaleta

15 de diciembre de 2016

Sitio web: www3.fi.mdp.edu.ar/fes/ProcQ.html

ALGORITMOS CUÁNTICOS AVANZADOS

Algoritmo de Factorización de SHOR y Búsqueda de Orden

Algoritmo de Factorización de SHOR

Descripción General

- Factorización Entera: La descomposición de un número compuesto en potencias de números primos.

Algoritmo de Factorización de SHOR

Descripción General

- Factorización Entera: La descomposición de un número compuesto en potencias de números primos.
- El problema real es cuando se requieren descomponer números grandes.

Algoritmo de Factorización de SHOR

Descripción General

- Factorización Entera: La descomposición de un número compuesto en potencias de números primos.
- El problema real es cuando se requieren descomponer números grandes.
- Objetivo: Factorizar en tiempo polinomial.

Algoritmo de Factorización de SHOR

Descripción General

- Factorización Entera: La descomposición de un número compuesto en potencias de números primos.
- El problema real es cuando se requieren descomponer números grandes.
- Objetivo: Factorizar en tiempo polinomial.
- Un algoritmo que pueda Factorizar un entero eficientemente podría comprometer seriamente la Criptografía RSA (Rivest, Shamir y Adleman).

Algoritmo de Factorización de SHOR

Descripción General

- Factorización Entera: La descomposición de un número compuesto en potencias de números primos.
- El problema real es cuando se requieren descomponer números grandes.
- Objetivo: Factorizar en tiempo polinomial.
- Un algoritmo que pueda Factorizar un entero eficientemente podría comprometer seriamente la Criptografía RSA (Rivest, Shamir y Adleman).
- RSA utiliza una clave pública N que es producto de dos números primos grandes.

Algoritmo de Factorización de SHOR

Descripción General

- Factorización Entera: La descomposición de un número compuesto en potencias de números primos.
- El problema real es cuando se requieren descomponer números grandes.
- Objetivo: Factorizar en tiempo polinomial.
- Un algoritmo que pueda Factorizar un entero eficientemente podría comprometer seriamente la Criptografía RSA (Rivest, Shamir y Adleman).
- RSA utiliza una clave pública N que es producto de dos números primos grandes.
- Actualmente $N \sim 2^{100}$

Algoritmo de Factorización de SHOR

Descripción General

- No existe un algoritmo clásico eficiente.

Algoritmo de Factorización de SHOR

Descripción General

- No existe un algoritmo clásico eficiente.
- El algoritmo de Shor puede factorizar y por lo tanto quebrar RSA en tiempo polinomial.

Algoritmo de Factorización de SHOR

Descripción General

- No existe un algoritmo clásico eficiente.
- El algoritmo de Shor puede factorizar y por lo tanto quebrar RSA en tiempo polinomial.
- Como la mayoría de los algoritmos cuánticos, el algoritmo de Shor es probabilístico.

Algoritmo de Factorización de SHOR

Descripción General

- No existe un algoritmo clásico eficiente.
- El algoritmo de Shor puede factorizar y por lo tanto quebrar RSA en tiempo polinomial.
- Como la mayoría de los algoritmos cuánticos, el algoritmo de Shor es probabilístico.
- Entrega una respuesta correcta con una alta probabilidad y la probabilidad de error decrece mediante la repetición del algoritmo.

Algoritmo de Factorización de SHOR

Descripción General

- No existe un algoritmo clásico eficiente.
- El algoritmo de Shor puede factorizar y por lo tanto quebrar RSA en tiempo polinomial.
- Como la mayoría de los algoritmos cuánticos, el algoritmo de Shor es probabilístico.
- Entrega una respuesta correcta con una alta probabilidad y la probabilidad de error decrece mediante la repetición del algoritmo.
- El algoritmo de Shor puede resolver en horas lo que a una computadora clásica le llevaría años.

La Búsqueda del Orden

La Búsqueda del Orden

Matemática Preliminar

- Los enteros *mod* N forman el conjunto $\{0, 1, 2, \cdots, N-1\}$ conocido como $\mathbb{Z}_N$.
- Dos enteros s y t son *equivalentes mod* N si N divide a $s - t$ exactamente. En este caso:

$$s \equiv t (mod\ N).$$

- Todo entero k puede ser *reducido mod* N tomando el resto r después de la división de k por N:

$$r = k\ mod\ N.$$

- Si $MCD(a, N) = 1$, el número 1 aparecerá en la secuencia $a\ mod\ N$, $a^2\ mod\ N$, $a^3\ mod\ N, \ldots$ y a partir de ahí la secuencia se repetirá en forma periódica.

La Búsqueda del Orden

Matemática Preliminar: Periodicidad

- De la teoría de números:
  $F(x) = a^x mod\ N$ es una función periódica.

Ejemplo: Elijamos $N = 15$ y $x = 7$

$$7^0 mod 15 = 1$$
$$7^1 mod 15 = 7$$
$$7^2 mod 15 = 4$$
$$7^3 mod 15 = 13$$
$$7^4 mod 15 = 1$$
$$\vdots$$

La Búsqueda del Orden

Ejemplo:

Potencias de 2:

$$2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, \ldots.$$

Potencias de 2 mod 15:

$$2, 4, 8, 1, 2, 4, 8, 1, 2, 4, 8, \ldots$$

La secuencia tiene **período** 4

Definición:

Dados los enteros a y N, que cumplen que $MCD(a, N) = 1$, el *orden* de $a \, mod \, N$ es el mínimo entero positivo r tal que $a^r \equiv 1 \, mod \, N$.

La Búsqueda del Orden

El Problema de Factorización de Enteros

- **Entrada:** Entero N.
- **Problema:** Encontrar los enteros positivos $p_1, p_2, \ldots, p_l, r_1, r_2, \ldots, r_l$
  - donde los p_i son primos distintos entre sí y $N = p_1^{r_1} p_2^{r_2} \ldots p_l^{r_l}$.

Supongamos que queremos factorizar el entero N

No necesitamos utilizar un algoritmo cuántico si:

1. N es par.
2. N no es primo.
3. N no tiene potencias de un único primo.

La Búsqueda del Orden

El Problema de Factorización de Enteros

- **Entrada:** Entero N.
- **Problema:** Encontrar los enteros positivos $p_1, p_2, \ldots, p_l, r_1, r_2, \ldots, r_l$
  - donde los p_i son primos distintos entre sí y $N = p_1^{r_1} p_2^{r_2} \ldots p_l^{r_l}$.

Supongamos que queremos factorizar el entero N

No necesitamos utilizar un algoritmo cuántico si:

1. N es par.
2. N no es primo.
3. N no tiene potencias de un único primo.

- Si podemos separar N en dos **factores no triviales**, entonces se puede encontrar los factores primos mediante algoritmos clásicos probabilísticos o deterministas de orden polinomial.

La Búsqueda del Orden

Supongamos que queremos factorizar el entero N

- El problema se puede reducir a la **separación de factores** en tiempos $O(log\ N)$.

Separación de Enteros

- **Entrada:** Entero impar N con por lo menos dos factores primos distintos.
- **Problema:** Encontrar dos enteros N_1 y N_2 tal que $N = N_1 \times N_2$.

La Búsqueda del Orden

Supongamos que queremos factorizar el entero N

- El problema se puede reducir a la **separación de factores** en tiempos $O(log\ N)$.

Separación de Enteros

- **Entrada:** Entero impar N con por lo menos dos factores primos distintos.
- **Problema:** Encontrar dos enteros N_1 y N_2 tal que $N = N_1 \times N_2$.
- Si encontramos un algoritmo eficiente para búsqueda del orden, es posible dar un algoritmo probabilístico eficiente para separar enteros.

La Búsqueda del Orden

Separar N

1. Encontrar el orden de un entero cualquiera **a** que sea *coprimo* de N.
2. Si a no es coprimo de N el $MCD(a, N)$ es un factor no trivial de N.
3. Para encontrar el MCD se usa el algoritmo extendido de Euclides. Se puede muestrear $\{2, 3, \ldots, N-2\}$ y testear con el algoritmo de Euclides para encontrar coprimos de N.
4. Si encontramos un coprimo a ($MCD(a, N) = 1$), el orden r de a será par con probabilidad $\frac{1}{2}$.
5. Si $b = a^{r/2} mod\ N \Rightarrow b^2 - 1 = 0 mod\ N$, y por lo tanto N divide a $(b-1)(b+1)$.
6. Se espera que $MCD(b-1, N)$ sea un factor no trivial de N.
7. Si N tiene dos factores primos distintos, para un a con orden r par, la probabilidad de que $MCD(a^{r/2} - 1 mod N, N)$ sea un factor no trivial de N es de $\frac{1}{2}$.

La Búsqueda del Orden

Búsqueda de Orden $\equiv$ buscar fracciones

- Sean enteros k seleccionados aleatoriamente de $\{0, 1, \ldots, r-1\}$
- **Se puede reducir el problema de búsqueda del orden a buscar fracciones $\frac{k}{r}$**

La Búsqueda del Orden

Búsqueda de Orden $\equiv$ buscar fracciones

- Sean enteros k seleccionados aleatoriamente de $\{0, 1, \ldots, r-1\}$
- **Se puede reducir el problema de búsqueda del orden a buscar fracciones $\frac{k}{r}$**

Teorema de Fracciones Continuas

- Todo número racional $\frac{x}{2^n}$ tiene una secuencia de $O(n)$ aproximaciones sucesivas, llamadas convergentes, $\frac{a_1}{b_1}, \frac{a_2}{b_2}, \ldots, \frac{a_m}{b_m}$, con $\frac{a_m}{b_m} = \frac{x}{2^n}$.
- La lista de convergentes de $\frac{x}{2^n}$ se calcula en tiempo polinomial en n.
- Este teorema muestra que se puede reducir la tarea de determinar exactamente la fracción $\frac{k}{r}$ a encontrar $\frac{x}{2^n}$ con $|\frac{x}{2^n} - \frac{k}{r}| \leq \frac{1}{2r^2}$.

La Búsqueda del Orden

Importante: Hasta acá algoritmos clásicos probabilísticos de orden polinomial.

Algoritmo de Fracciones Continuas

El algoritmo de fracciones continuas es un método para encontrar las aproximaciones sucesivas a un número real.

Ejemplo: Descomponer 31/13 en fracciones continuas

- Separar 31/13 en partes entera y fraccionaria. $\frac{31}{13} = 2 + \frac{5}{13}$.
- invertimos la parte fraccionaria: $\frac{31}{13} = 2 + \frac{1}{\frac{13}{5}}$.

Ejemplo: Descomponer 31/13 en fracciones continuas (sigue)

- Los pasos de separar e invertir son aplicados a 13/5.

$$\frac{31}{13} = 2 + \frac{1}{2 + \frac{3}{5}} = 2 + \frac{1}{2 + \frac{1}{\frac{5}{3}}}.$$

- Separar e invertir 5/3: $\frac{31}{13} = 2 + \frac{1}{2+\frac{1}{1+\frac{2}{3}}} = 2 + \frac{1}{2+\frac{1}{1+\frac{1}{\frac{3}{2}}}}$.
- $\frac{31}{13} = 2 + \frac{1}{2+\frac{1}{1+\frac{1}{1+\frac{1}{2}}}}$.

- El algoritmo termina después de un número finito de pasos para un número racional.
- En resumen: La búsqueda de orden r de a modulo N se reduce al problema de estimaciones de muestreo.

La Búsqueda del Orden

Estimaciones de Muestreo de un entero aleatorio múltiplo de $\frac{1}{r}$

- **Entrada:** Enteros a y N tales que $MCD(a, N) = 1$. Sea r el orden (desconocido) de a.
- **Problema:** Obtener $x \in \{0, 1, 2, \ldots, 2^{n-1}\}$
  - tal que para $k \in \{0, 1, \ldots, r-1\}$ se tiene $Pr\left(|\frac{x}{2^n} - \frac{k}{r}| \leq \frac{1}{2r^2}\right)$

La Búsqueda del Orden

Estimaciones de Muestreo de un entero aleatorio múltiplo de $\frac{1}{r}$

- **Entrada:** Enteros a y N tales que $MCD(a, N) = 1$. Sea r el orden (desconocido) de a.
- **Problema:** Obtener $x \in \{0, 1, 2, \ldots, 2^{n-1}\}$
  - tal que para $k \in \{0, 1, \ldots, r-1\}$ se tiene $Pr\left(|\frac{x}{2^n} - \frac{k}{r}| \leq \frac{1}{2r^2}\right)$

En el problema de muestreo es donde se usa un algoritmo cuántico.

La Búsqueda del Orden

Estimaciones de Muestreo de un entero aleatorio múltiplo de $\frac{1}{r}$

- **Entrada:** Enteros a y N tales que $MCD(a, N) = 1$. Sea r el orden (desconocido) de a.
- **Problema:** Obtener $x \in \{0, 1, 2, \ldots, 2^{n-1}\}$
  - tal que para $k \in \{0, 1, \ldots, r-1\}$ se tiene $Pr\left(|\frac{x}{2^n} - \frac{k}{r}| \leq \frac{1}{2r^2}\right)$

En el problema de muestreo es donde se usa un algoritmo cuántico.

Estimación de Autovalores para la Búsqueda de Orden

- Sea $U_a : |s\rangle \rightarrow |sa \bmod N\rangle, \quad 0 \leq s < N.$
- Dado que $a^r \equiv 1 (mod\ N)$, tenemos $U_a^r : |s\rangle \rightarrow |sa^r \bmod N\rangle = |s\rangle$
- O sea que U_a es una raíz r-ésima de la identidad.
- $amodN$ tiene orden $r \Rightarrow U_a$ es una raíz r-ésima de la identidad.

Ejercicio: Pruebe que si un operador U cumple que $U^r = I$, entonces sus autovalores deben ser raíces r-ésima de 1, o sea de la forma $e^{2\pi i \frac{k}{r}}$ para algún entero k

Consideremos el estado $|u_k\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} |a^s \, mod N\rangle$.

$$\begin{aligned} U_a|u_k\rangle &= \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} U_a |a^s \, mod N\rangle \\ &= \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} \left|a^{s+1} \, mod N\right\rangle \\ &= e^{2\pi i \frac{k}{r}} \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r}(s+1)} \left|a^{s+1} \, mod N\right\rangle \\ &= e^{2\pi i \frac{k}{r}} |u_k\rangle \end{aligned}$$

Por lo tanto $|u_k\rangle$ es una autoestado de U_a con autovalor $e^{2\pi i \frac{k}{r}}$

Ejercicio: Pruebe que si un operador U cumple que $U^r = I$, entonces sus autovalores deben ser raíces r-ésima de 1, o sea de la forma $e^{2\pi i \frac{k}{r}}$ para algún entero k

Consideremos el estado $|u_k\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} |a^s \, mod N\rangle$.

$$\begin{aligned} U_a|u_k\rangle &= e^{2\pi i \frac{k}{r}} \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r}(s+1)} \left|a^{s+1} \, mod N\right\rangle \\ &= e^{2\pi i \frac{k}{r}} |u_k\rangle \end{aligned}$$

$e^{2\pi i \frac{k}{r} r} |a^r \, mod N\rangle = e^{2\pi i \frac{k}{r} 0} \left|a^0 \, mod N\right\rangle$.

La Búsqueda del Orden

Ejercicio: Pruebe que si un operador U cumple que $U^r = I$, entonces sus autovalores deben ser raíces r-ésima de 1, o sea de la forma $e^{2\pi i \frac{k}{r}}$ para algún entero k

Consideremos el estado $|u_k\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} |a^s \, mod N\rangle$.

$$\begin{aligned} U_a|u_k\rangle &= e^{2\pi i \frac{k}{r}} \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r}(s+1)} \left|a^{s+1} \, mod N\right\rangle \\ &= e^{2\pi i \frac{k}{r}} |u_k\rangle \end{aligned}$$

$e^{2\pi i \frac{k}{r} r} |a^r \, mod N\rangle = e^{2\pi i \frac{k}{r} 0} \left|a^0 \, mod N\right\rangle$.

Para cualquier valor de k entre 0 y $r-1$, podemos aplicar el algoritmo de estimación de autovalores para obtener k/r y así resolver el problema de búsqueda del orden.

¿Cómo preparamos $|u_k\rangle$ sin conocer r?

- No es necesario conocer r.
- El algoritmo de estimación de autovalores produce una superposición de estos autoestados *entangled* con las estimaciones de sus autovalores y una medición producirá como resultado una estimación de alguno de los autovalores.

Tomemos

$$\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|u_k\rangle = \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}e^{-2\pi i\frac{k}{r}s}U_a|a^s \, mod \, N\rangle.$$

Si $s = 0(mod\, r)$ entonces $|a^s \, mod \, N\rangle = |1\rangle$. La amplitud de $|1\rangle$ en el estado de arriba es entonces la suma de las amplitudes sobre los términos con $s = 0$, o sea

$$\begin{aligned}\frac{1}{\sqrt{r}}\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}e^{-2\pi i\frac{k}{r}0} &= \frac{1}{r}\sum_{k=0}^{r-1}1 \\ &= 1.\end{aligned}$$

¿Cómo preparamos $|u_k\rangle$ sin conocer r?

- Por lo tanto la amplitud del estado $|1\rangle$ es 1 y la amplitud del resto de los vectores de la base debe ser 0. Se tiene entonces

$$\frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |u_k\rangle = |1\rangle.$$

- Ésto significa que el algoritmo de estimación de autovalores transforma el estado de entrada

$$\begin{aligned} |0\rangle|1\rangle &= |0\rangle\left(\tfrac{1}{\sqrt{r}} \textstyle\sum_{k=0}^{r-1} |u_k\rangle\right) \\ &= \tfrac{1}{\sqrt{r}} \textstyle\sum_{k=0}^{r-1} |0\rangle|u_k\rangle \end{aligned}$$

al estado

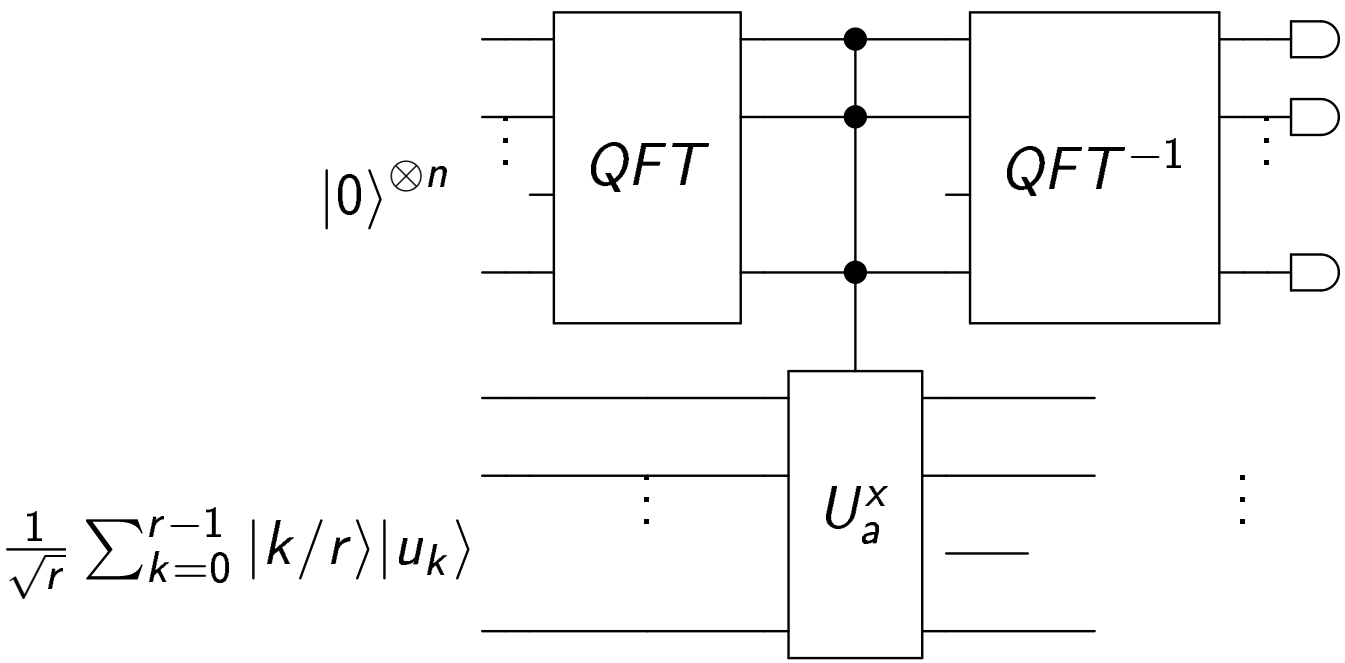
$$\frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |k/r\rangle|u_k\rangle.$$

¿Cómo preparamos $|u_k\rangle$ sin conocer r?

Conclusión:

- Como el primer registro está en una mezcla uniforme de estados $|k/r\rangle$, una medición del mismo producirá un entero x, tal que $\frac{x}{2^n}$ es una estimación de $\frac{k}{r}$ para algún $k \in \{0, 1, \ldots, r-1\}$.
- Esta estimación nos permite determinar con probabilidad alta $\frac{k}{r}$ de acuerdo al teorema de fracciones continuas.

Circuito para la estimación de un múltiplo de $\frac{1}{r}$



Factorización de $N = 15$

- Primero elegimos un número que no tenga factores comunes con 15, por ejemplo $x = 7$
- Algoritmo cuántico para encontrar el orden r de x respecto de N.
- Se comienza con el estado $|0\rangle|0\rangle$ y aplicando QFT al primer registro se obtiene:

$$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle|0\rangle = \frac{1}{\sqrt{2^n}} \Big[|0\rangle + |1\rangle + \cdots + |2^n - 1\rangle \Big] |0\rangle$$

El valor de $n = 11$ es elegido para tener una probabilidad de error de 1/4 como máximo.

- Luego se calcula $x^k \, mod N$ ubicando el resultado en el segundo registro (Operación U_a^x controlada sobre el bit target del circuito cuántico)

Factorización de $N = 15$

- Así se obtiene

$$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \left|x^k \, mod N\right\rangle$$

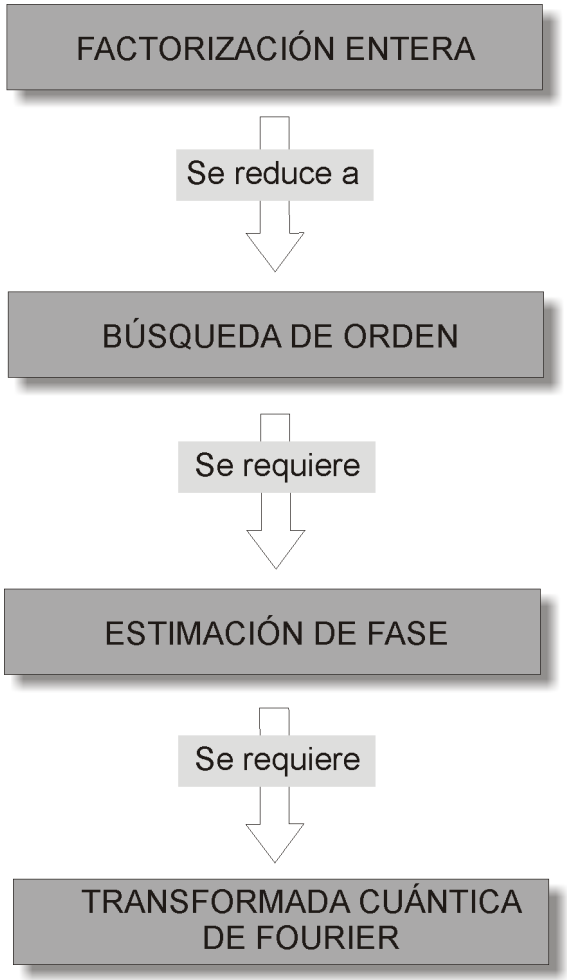
$$= \frac{1}{\sqrt{2^n}} \Big[|0\rangle|1\rangle + |1\rangle|7\rangle + |2\rangle|4\rangle + |3\rangle|13\rangle + \\ + |4\rangle|1\rangle + |5\rangle|7\rangle + |6\rangle|4\rangle + \ldots \Big].$$

- QFT^{-1} al primer registro y calcular k/r para sacar r de ahí. (Observemos QUE el segundo registro tiene una periodicidad y repite 1, 7, 4 y 13)
- El estado del primer registro al que se aplica QFT^{-1} es $\sqrt{\frac{4}{2^n}} \Big[|2\rangle + |6\rangle + |10\rangle + |14\rangle + \ldots \Big]$.
- Después de aplicar QFT^{-1} se obtiene $\sum_l \alpha_l |l\rangle$, $l = 2^n = 2048$.
- La distribución de probabilidad de este estado tiene cuatro picos de 1/4 cada uno en 0, 512, 1024 y 1536.

Factorización de $N = 15$

- Suponemos que se mide 1536.
- $1536/2048 = 3/4 = 1/(1 + 1/3)$, y $3/4$ aparece como convergente en la expansión dando $r = 4$ como orden de $x = 7$ con $N = 15$.
- Cuando el orden es par, como en este caso, $b = x^{r/2} mod N$ y $b^2 - 1 = 0 mod N$ y se cumple que N divide a $(b - 1)(b + 1)$.
- Se prueba con $MCD(b - 1, N)$ o $MCD(b + 1, N)$ para encontrar un factor no trivial de N, en este caso $b - 1 = 3$ y $b + 1 = 5$.

Factorización Entera, búsqueda de orden, la estimación de fase y la QFT



Algoritmo Original de Shor para Búsqueda de Orden

- La diferencia entre los dos algoritmos es la base en la que el estado del segundo registro es expresado: en la versión previa en la base de autovectores y en la de Shor en la computacional.

	Shor	Estimación Autovalores
Estado Inicial	$\|0\rangle\|1\rangle$	$\sum_k \|0\rangle\|u_k\rangle$
QFT	$\sum_x \|x\rangle\|1\rangle$	$\sum_k \sum_x \|x\rangle\|u_k\rangle$
$c - U_a^x$	$\sum_b (\sum_z \|zr + b\rangle)\|a^b\rangle$	$\sum_k \left(\sum_x e^{2\pi i \frac{kx}{r}} \|x\rangle\right)\|u_k\rangle$
QFT^{-1}	$\frac{x}{2^n}$	$\frac{x}{2^n}$