

Introducción a la Computación Cuántica, Algoritmos e Implementación

Chenjie Huang

Sección Departamental de Sistemas Informáticos y Computación, Tutor: Luis Fernando Llana Díaz



UNIVERSIDAD
COMPLUTENSE
MADRID

Índice

- 1 Quantum Bit
- 2 Puertas cuánticas
- 3 Algoritmos cuánticos
- 4 Ejecuciones de algoritmos
- 5 Conclusiones

Bit Cúantico (Qubit)

Quantum Bit

- Puede tomar dos estados clásicos $|0\rangle$ y $|1\rangle$.

Bit Cúántico (Qubit)

Quantum Bit

- Puede tomar dos estados clásicos $|0\rangle$ y $|1\rangle$.
- O cualquier combinación de ellas:

$$|\varphi\rangle = \alpha |0\rangle + \beta |1\rangle \quad \alpha, \beta \in \mathbb{C} \quad (1)$$

donde $|\alpha|^2 + |\beta|^2 = 1$.

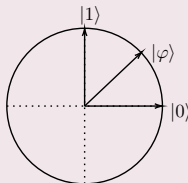
Bit Cúántico (Qubit)

Quantum Bit

- Puede tomar dos estados clásicos $|0\rangle$ y $|1\rangle$.
- O cualquier combinación de ellas:

$$|\varphi\rangle = \alpha |0\rangle + \beta |1\rangle \quad \alpha, \beta \in \mathbb{C} \quad (1)$$

donde $|\alpha|^2 + |\beta|^2 = 1$.



Bit Cúantico (Qubit)

Esfera de Bloch

Otra forma de representarlo es con coordenadas polares en una esfera.

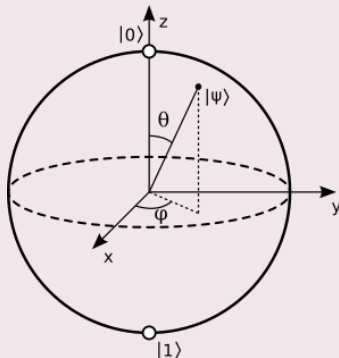
$$|\psi\rangle = \cos \theta |0\rangle + e^{i\varphi} \sin \theta |1\rangle \quad (2)$$

Bit Cúántico (Qubit)

Esfera de Bloch

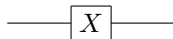
Otra forma de representarlo es con coordenadas polares en una esfera.

$$|\psi\rangle = \cos \theta |0\rangle + e^{i\varphi} \sin \theta |1\rangle \quad (2)$$



Puertas Cuánticas

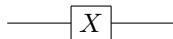
Puerta *NOT* (Pauli *X*):



$$\begin{cases} |0\rangle \rightarrow |1\rangle \\ |1\rangle \rightarrow |0\rangle \end{cases} \quad (3)$$

Puertas Cuánticas

Puerta *NOT* (Pauli X):



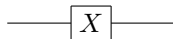
$$\begin{cases} |0\rangle \rightarrow |1\rangle \\ |1\rangle \rightarrow |0\rangle \end{cases} \quad (3)$$

Representación Matricial:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (4)$$

Puertas Cuánticas

Puerta *NOT* (Pauli X):



$$\begin{cases} |0\rangle \rightarrow |1\rangle \\ |1\rangle \rightarrow |0\rangle \end{cases} \quad (3)$$

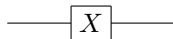
Representación Matricial:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (5)$$

Puertas Cuánticas

Puerta *NOT* (Pauli X):



$$\begin{cases} |0\rangle \rightarrow |1\rangle \\ |1\rangle \rightarrow |0\rangle \end{cases} \quad (3)$$

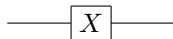
Representación Matricial:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} |0\rangle = |1\rangle \quad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (5)$$

Puertas Cuánticas

Puerta *NOT* (Pauli X):



$$\begin{cases} |0\rangle \rightarrow |1\rangle \\ |1\rangle \rightarrow |0\rangle \end{cases} \quad (3)$$

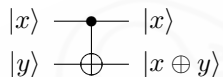
Representación Matricial:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} |0\rangle = |1\rangle \quad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} |1\rangle = |0\rangle \quad (5)$$

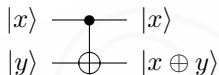
Puertas Cuánticas

Puerta controlada *NOT* (*CNOT*):



Puertas Cuánticas

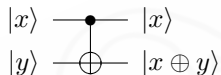
Puerta controlada *NOT* (*CNOT*):



$$\begin{cases} |0\rangle \otimes |0\rangle \rightarrow |0\rangle \otimes |0\rangle \\ |0\rangle \otimes |1\rangle \rightarrow |0\rangle \otimes |1\rangle \end{cases} \quad \begin{cases} |1\rangle \otimes |0\rangle \rightarrow |1\rangle \otimes |1\rangle \\ |1\rangle \otimes |1\rangle \rightarrow |1\rangle \otimes |0\rangle \end{cases} \quad (6)$$

Puertas Cuánticas

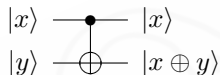
Puerta controlada *NOT* (*CNOT*):



$$\left\{ \begin{array}{l} |00\rangle \rightarrow |00\rangle \\ |01\rangle \rightarrow |01\rangle \end{array} \right. \quad \left\{ \begin{array}{l} |10\rangle \rightarrow |11\rangle \\ |11\rangle \rightarrow |10\rangle \end{array} \right. \quad (6)$$

Puertas Cuánticas

Puerta controlada *NOT* (*CNOT*):

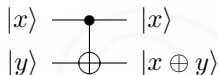


$$\begin{cases} |00\rangle \rightarrow |00\rangle \\ |01\rangle \rightarrow |01\rangle \end{cases} \quad \begin{cases} |10\rangle \rightarrow |11\rangle \\ |11\rangle \rightarrow |10\rangle \end{cases} \quad (6)$$

$$CNOT |10\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (7)$$

Puertas Cuánticas

Puerta controlada *NOT* (*CNOT*):

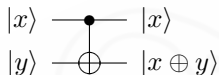


$$\begin{cases} |00\rangle \rightarrow |00\rangle \\ |01\rangle \rightarrow |01\rangle \end{cases} \quad \begin{cases} |10\rangle \rightarrow |11\rangle \\ |11\rangle \rightarrow |10\rangle \end{cases} \quad (6)$$

$$CNOT|10\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (7)$$

Puertas Cuánticas

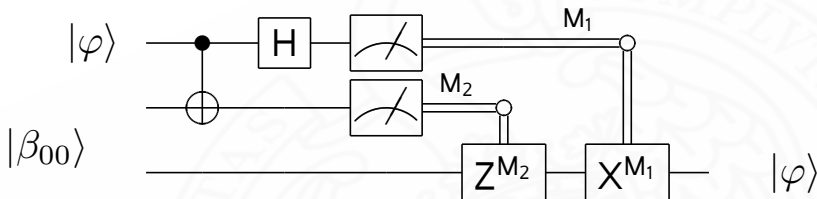
Puerta controlada *NOT* (*CNOT*):



$$\begin{cases} |00\rangle \rightarrow |00\rangle \\ |01\rangle \rightarrow |01\rangle \end{cases} \quad \begin{cases} |10\rangle \rightarrow |11\rangle \\ |11\rangle \rightarrow |10\rangle \end{cases} \quad (6)$$

$$CNOT |10\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} |1\rangle \otimes |0\rangle = |1\rangle \otimes |1\rangle \quad (7)$$

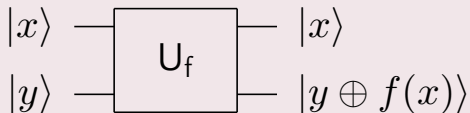
Algoritmo de Teletransportación



Algoritmo de Deutsch

Oráculo

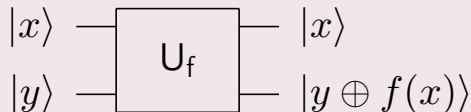
El oráculo es una operación unitaria U_f que nos permitirá evaluar una función $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$.



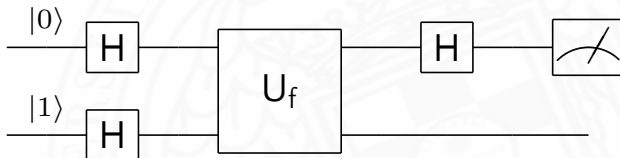
Algoritmo de Deutsch

Oráculo

El oráculo es una operación unitaria U_f que nos permitirá evaluar una función $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$.



Circuito para el algoritmo de Deutsch:

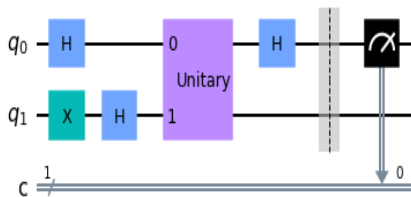


Algoritmo de Deutsch

```

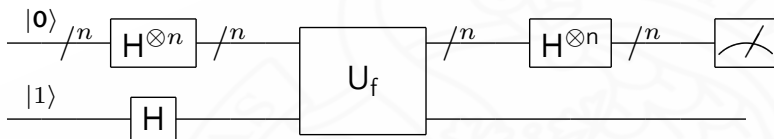
circ = QuantumCircuit(2, 1)
circ.x(1)
circ.h(range(2))
gf = Operator([[0, 0, 1, 0],
               [0, 1, 0, 0],
               [1, 0, 0, 0],
               [0, 0, 0, 1]])
circ.append(gf, [0, 1])
circ.h(0)
circ.barrier(range(2))
circ.measure(range(1), range(1))
circ.draw('mpl')

```



Algoritmo de Deutsch-Jozsa

Circuito para el algoritmo de Deutsch-Jozsa:

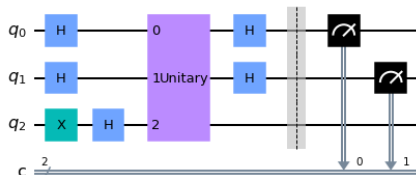


Algoritmo de Deutsch-Jozsa

```

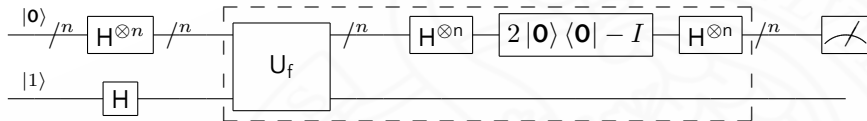
circ = QuantumCircuit(3,2)
circ.x(2)
circ.h(range(3))
gf = Operator(
    [[1, 0, 0, 0, 0, 0, 0, 0],
     [0, 0, 0, 0, 0, 1, 0, 0],
     [0, 0, 1, 0, 0, 0, 0, 0],
     [0, 0, 0, 0, 0, 0, 0, 1],
     [0, 0, 0, 0, 1, 0, 0, 0],
     [0, 1, 0, 0, 0, 0, 0, 0],
     [0, 0, 0, 0, 0, 0, 1, 0],
     [0, 0, 0, 1, 0, 0, 0, 0]])
circ.append(gf, [0, 1, 2])
circ.h(range(2))
circ.barrier(range(3))
circ.measure(range(2), range(2))
circ.draw('mpl')

```



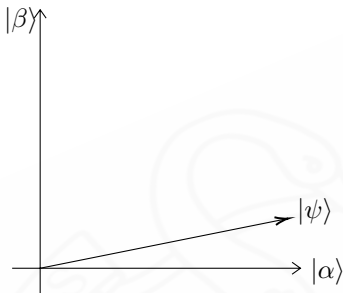
Algoritmo de Búsqueda de Grover

Circuito para el algoritmo de Grover:



Algoritmo de Búsqueda de Grover

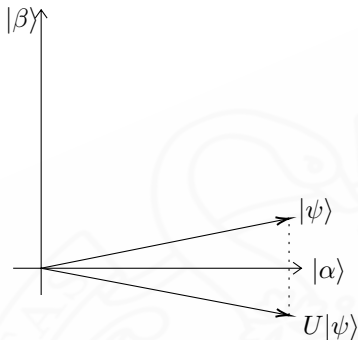
Representación geométrica:



$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{\mathbf{x} \in \{0,1\}^n} |\mathbf{x}\rangle \quad |\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_{f(\mathbf{x})=0} |\mathbf{x}\rangle \quad |\beta\rangle = \frac{1}{\sqrt{M}} \sum_{f(\mathbf{x})=1} |\mathbf{x}\rangle \quad (8)$$

Algoritmo de Búsqueda de Grover

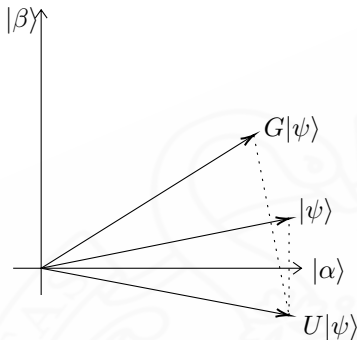
Representación geométrica:



$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{\mathbf{x} \in \{0,1\}^n} |\mathbf{x}\rangle \quad |\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_{f(\mathbf{x})=0} |\mathbf{x}\rangle \quad |\beta\rangle = \frac{1}{\sqrt{M}} \sum_{f(\mathbf{x})=1} |\mathbf{x}\rangle \quad (8)$$

Algoritmo de Búsqueda de Grover

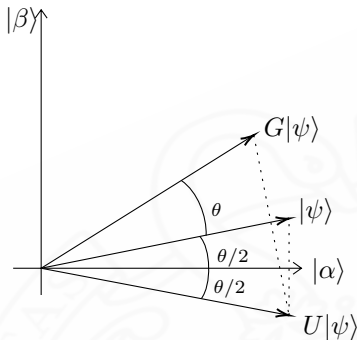
Representación geométrica:



$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{\mathbf{x} \in \{0,1\}^n} |\mathbf{x}\rangle \quad |\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_{f(\mathbf{x})=0} |\mathbf{x}\rangle \quad |\beta\rangle = \frac{1}{\sqrt{M}} \sum_{f(\mathbf{x})=1} |\mathbf{x}\rangle \quad (8)$$

Algoritmo de Búsqueda de Grover

Representación geométrica:



$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{\mathbf{x} \in \{0,1\}^n} |\mathbf{x}\rangle$$

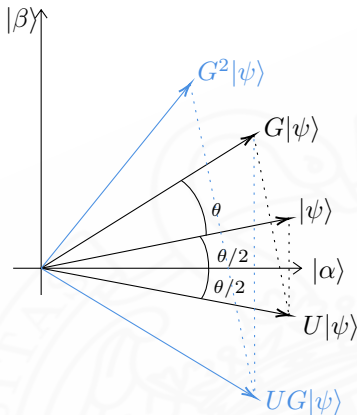
$$|\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_{f(\mathbf{x})=0} |\mathbf{x}\rangle$$

$$|\beta\rangle = \frac{1}{\sqrt{M}} \sum_{f(\mathbf{x})=1} |\mathbf{x}\rangle$$

(8)

Algoritmo de Búsqueda de Grover

Representación geométrica:



$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{\mathbf{x} \in \{0,1\}^n} |\mathbf{x}\rangle$$

$$|\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_{f(\mathbf{x})=0} |\mathbf{x}\rangle$$

$$|\beta\rangle = \frac{1}{\sqrt{M}} \sum_{f(\mathbf{x})=1} |\mathbf{x}\rangle$$

(8)

Algoritmo de Periodicidad de Simon

Periodicidad

Sea $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ tal que se cumple que existe una cadena binaria $\mathbf{c} \in \{0, 1\}^n$, para todo $\mathbf{x}, \mathbf{y} \in \{0, 1\}^n$ se cumple

$$f(\mathbf{x}) = f(\mathbf{y}) \text{ si y sólo si } \mathbf{x} = \mathbf{y} \oplus \mathbf{c} \quad (9)$$

donde \oplus es la suma módulo 2 dígito a dígito. Llamaremos entonces a \mathbf{c} el periodo de la función.

Algoritmo de Periodicidad de Simon

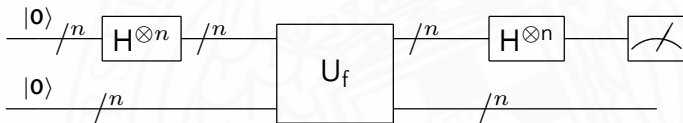
Periodicidad

Sea $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ tal que se cumple que existe una cadena binaria $\mathbf{c} \in \{0, 1\}^n$, para todo $\mathbf{x}, \mathbf{y} \in \{0, 1\}^n$ se cumple

$$f(\mathbf{x}) = f(\mathbf{y}) \text{ si y sólo si } \mathbf{x} = \mathbf{y} \oplus \mathbf{c} \quad (9)$$

donde \oplus es la suma módulo 2 dígito a dígito. Llamaremos entonces a \mathbf{c} el periodo de la función.

Circuito del algoritmo de Simon:



Algoritmo de Periodicidad de Simon

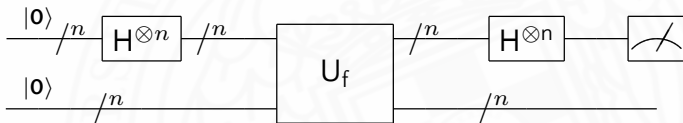
Periodicidad

Sea $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ tal que se cumple que existe una cadena binaria $\mathbf{c} \in \{0, 1\}^n$, para todo $\mathbf{x}, \mathbf{y} \in \{0, 1\}^n$ se cumple

$$f(\mathbf{x}) = f(\mathbf{y}) \text{ si y sólo si } \mathbf{x} = \mathbf{y} \oplus \mathbf{c} \quad (9)$$

donde \oplus es la suma módulo 2 dígito a dígito. Llamaremos entonces a \mathbf{c} el periodo de la función.

Circuito del algoritmo de Simon:



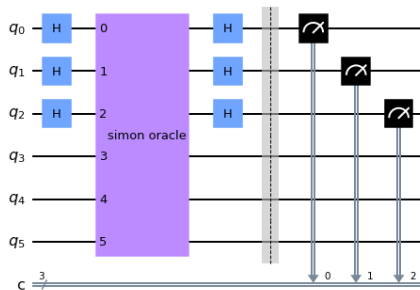
Se obtiene en la medición las cadenas \mathbf{z}_i que cumplen que $\langle \mathbf{z}_i, \mathbf{c} \rangle = 0$.

Algoritmo de Periodicidad de Simon

```

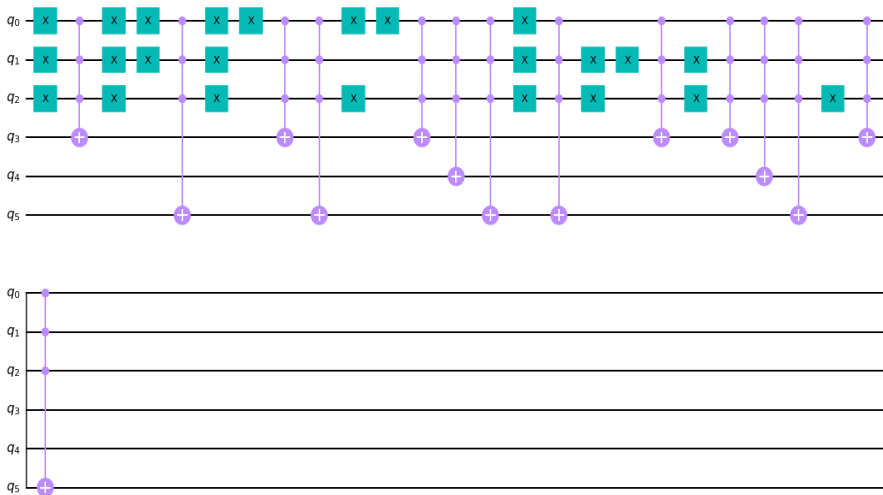
qc = QuantumCircuit(6,3)
qc.h(range(3))
qc.append(simon_oracle ,
          range(6))
qc.h(range(3))
qc.barrier()
qc.measure(range(3), range(3))
qc.draw('mpl')

```



Algoritmo de Periodicidad de Simon

Oráculo del circuito:



Algoritmo de Factorización de Shor

- Aplicaremos primero un procedimiento clásico para el algoritmo.



Algoritmo de Factorización de Shor

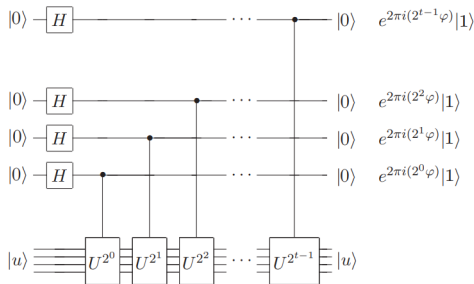
- Aplicaremos primero un procedimiento clásico para el algoritmo.
- Aplicaremos después un procedimiento cuántico para el algoritmo. Lo usaremos para encontrar el orden r de $x \bmod N$.



Algoritmo de Factorización de Shor

- Aplicaremos primero un procedimiento clásico para el algoritmo.
- Aplicaremos después un procedimiento cuántico para el algoritmo. Lo usaremos para encontrar el orden r de $x \bmod N$.

Estimación de Fase (Quantum Phase Estimation)

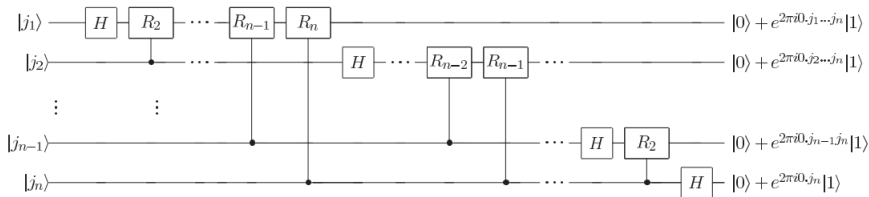


El operador unitario que usaremos para hallar el orden es

$$U |y\rangle = |xy \bmod N\rangle \quad (10)$$

Algoritmo de Factorización de Shor

Transformada Cuántica de Fourier (Quantum Fourier Transformation)



Donde R_k es la rotación que tiene por matriz:

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i / 2^k} \end{bmatrix} \quad (11)$$

Ejecuciones de algoritmos

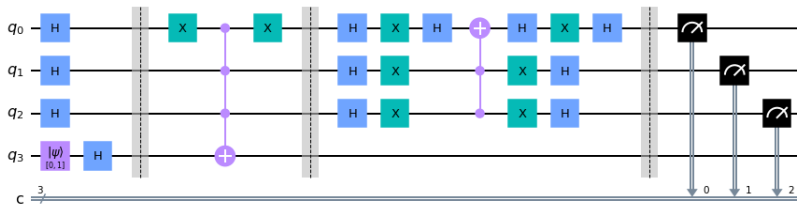
Código en qiskit para el algoritmo de Grover:

```
#circuito_2
#solucion esperado: 110
gc = QuantumCircuit(4,3)
gc.h([0,1,2])
gc.initialize([0, 1], 3)
gc.h(3)
gc.barrier(range(4))
#iteración de grover
#oraculo
gc.x(0)
gc.mcx([0, 1, 2], 3)
gc.x(0)
gc.barrier(range(4))

#segunda simetría
gc.h([0,1,2])
gc.x([0,1,2])
gc.h(0)
gc.ccx(1,2,0)
gc.h(0)
gc.x([0,1,2])
gc.h([0,1,2])
gc.barrier(range(4))
gc.measure(range(3), range(3))
gc.draw('mpl')
```

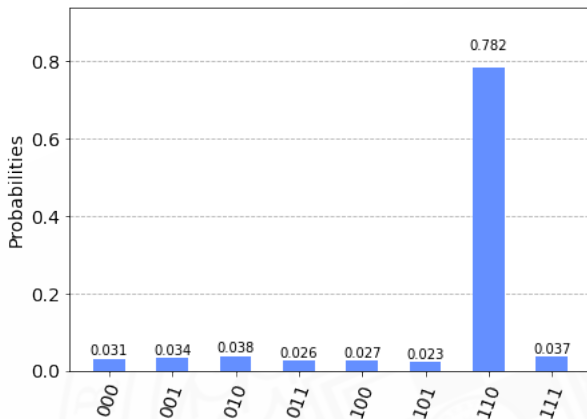

Ejecuciones de algoritmos

Circuito cuántico de Grover implementado en qiskit:



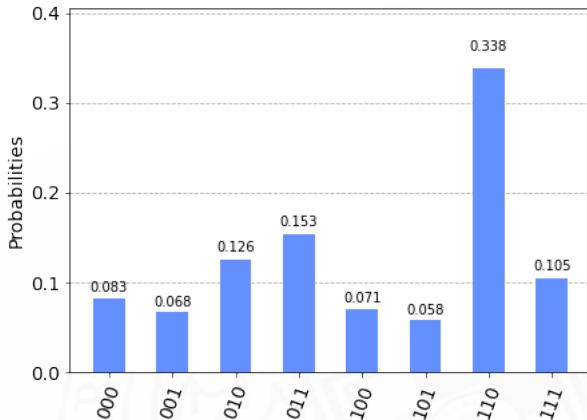
Ejecuciones de algoritmos

Simulación en un ordenador clásico:



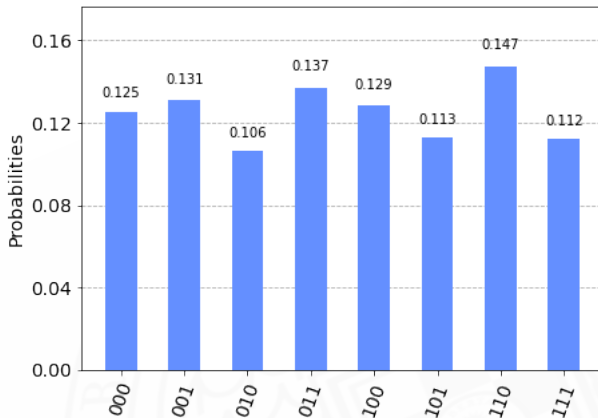
Ejecuciones de algoritmos

Ejecución en un ordenador cuántico:



Ejecuciones de algoritmos

Ejecución en un ordenador cuántico:



Conclusiones

- Se requiere conocimientos sobre álgebra lineal y producto tensorial para la computación cuántica.

Conclusiones

- Se requiere conocimientos sobre álgebra lineal y producto tensorial para la computación cuántica.
- A parte del área de estudio de cada algoritmo. Como el caso del algoritmo de factorización de Shor con la teoría de números.

Conclusiones

- Se requiere conocimientos sobre álgebra lineal y producto tensorial para la computación cuántica.
- A parte del área de estudio de cada algoritmo. Como el caso del algoritmo de factorización de Shor con la teoría de números.
- Hay temas más complejos como la existencia de puertas universales.

Conclusiones

- Se requiere conocimientos sobre álgebra lineal y producto tensorial para la computación cuántica.
- A parte del área de estudio de cada algoritmo. Como el caso del algoritmo de factorización de Shor con la teoría de números.
- Hay temas más complejos como la existencia de puertas universales.
- Las implementaciones se han hecho con un número bajo de qubits.

Conclusiones

- Se requiere conocimientos sobre álgebra lineal y producto tensorial para la computación cuántica.
- A parte del área de estudio de cada algoritmo. Como el caso del algoritmo de factorización de Shor con la teoría de números.
- Hay temas más complejos como la existencia de puertas universales.
- Las implementaciones se han hecho con un número bajo de qubits.
- La tecnología de hoy en día no permite una implementación totalmente funcional.