

Universidad Nacional del Altiplano

Facultad de Ingeniería Estadística e Informática

Docente: Fred Torres Cruz

Alumno: Roberto Angel Ticona Miramira

Algoritmo de Grover

Enlace al repositorio de Github: Algoritmo de Grover

Definición

El algoritmo de Grover es uno de los principales algoritmos de la computación cuántica y es uno de los más básicos, el cual explota al máximo el principio fundamental de superposición, mostrando la superioridad de las computadoras cuánticas sobre las clásicas. Este algoritmo se conoce por su poder de búsqueda sobre un conjunto de datos no estructurados, es decir imaginemos que tenemos un conjunto con N datos de estudiantes, pero necesitamos encontrar uno en específico el cual para este algoritmo lo llamaremos target (w), si realizamos la acción de búsqueda sobre estos N estudiantes, con un algoritmo clásico debería mínimo buscarlo en $N/2$ operaciones. En un sistema cuántico dada la superposición de estados, este problema se puede resolver examinando simultáneamente todas las posibles combinaciones. Como resultado, el número de pasos para encontrar el nombre del estudiante que deseamos se puede obtener en solo $O(\sqrt{N})$

Como funciona

Una de las cosas muy importantes es un computador cuántico es saber la forma en la que ingresan elementos de la lista, para la implementación de este algoritmo se establece una función f , la cual tiene como objetivo marcar el estado como $f(x) = 0$ si el elemento que estamos buscando no es w (target) y si cambiamos por el estado marcado, entonces $f(w) = 1$ es el marcado, al decodificar la función en una matriz unitaria esta se llamara el oráculo U_f . Una vez establecido el comportamiento de la función, si la observamos a manera de qubits obtenemos que $f : \{0, 1\}^n \rightarrow \{0, 1\}$ dado que $N = 2^n$ por el número de qubits que necesitamos para la representación del número N .

$$f(x) = \begin{cases} 1 & x = x_0 \\ 0 & x \neq x_0 \end{cases} \quad \text{El objetivo de la función es hallar } x_0$$

Para definir la matriz U_f (oráculo), tenemos que tener en cuenta que esta debe actuar sobre todos los estados $|x\rangle$. Esta matriz está definida de la siguiente manera:

$$U_f|x\rangle = (-1)^{f(x)}|x\rangle$$

Como podemos observar en la función anterior, si el estado x no es el marcado, no tendrá efecto sobre el estado $|x\rangle$. Ahora bien, si $U_f|w\rangle = (-1)^{f(w)}|w\rangle$, el resultado será:

$$U_f|w\rangle = -|w\rangle.$$

Geométricamente, la matriz unitaria corresponde a hacer una reflexión en su amplitud sobre el estado marcado dentro del conjunto de N elementos. Con la función anterior, podemos marcar el estado que deseamos buscar, y este modifica la amplitud del estado para que resalte sobre los demás estados.

Como lo mencionamos anteriormente, este algoritmo funciona utilizando la superposición, dado que no conocemos dónde está el artículo marcado dentro del conjunto de N elementos. Por lo tanto, utilizamos la definición matemática de la superposición uniforme aplicada a los estados cuánticos.

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

Esta superposición permite que, por medio del procedimiento anterior de amplificación de amplitud, que es como el computador cuántico aumento significativamente esta probabilidad. Este procedimiento cuando amplifica la amplitud del elemento marcado reduce la amplitud de los otros elementos, por lo que la medición en el estado final devolverá al elemento correcto casi con certeza y con una probabilidad casi de 100 % del estado marcado.

Este algoritmo trabaja en 3 pasos. El primero de estos es la inicialización de las amplitudes de los estados, las cuales corresponden al hacer superposición $|s\rangle$. Esta superposición se expresa por medio de compuertas cuánticas como:

En el instante $t = 0$, tenemos que:

$$|\psi_t\rangle = |s\rangle.$$

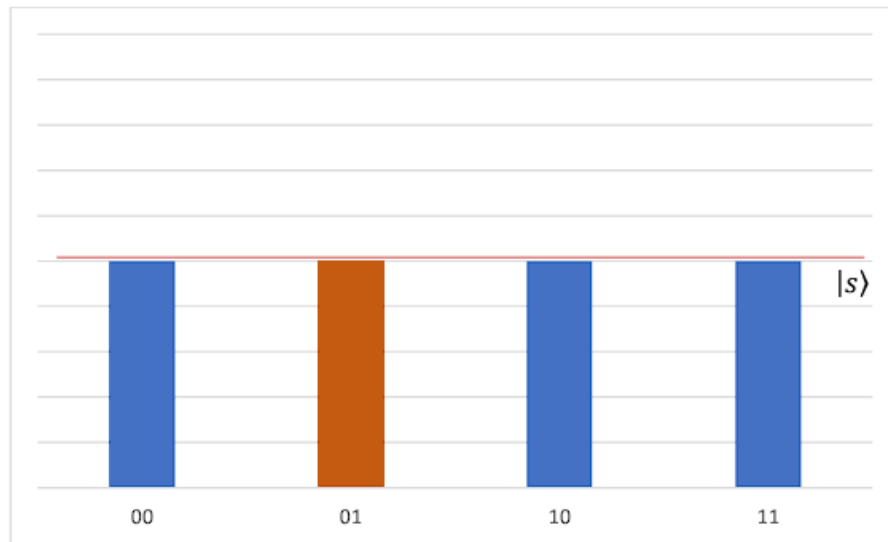
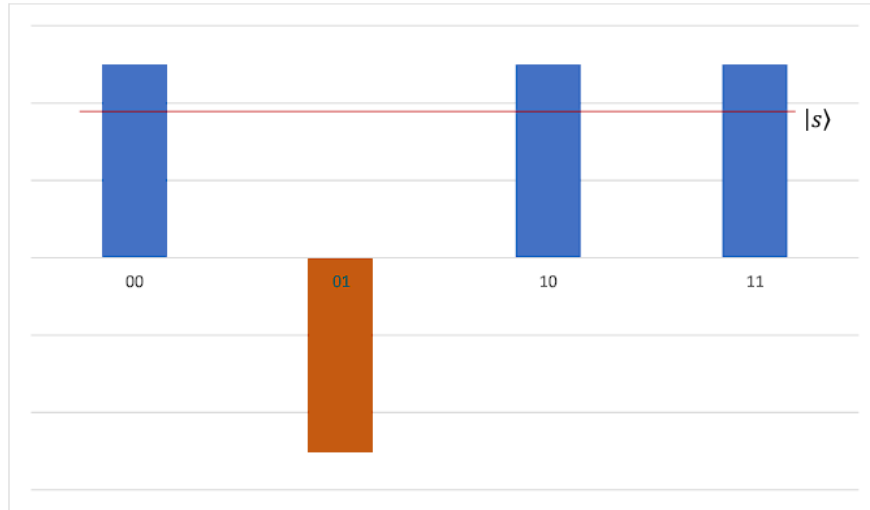


Figura 1: Amplitudes iniciales paso 1 Grover

En el segundo paso, se aplica la reflexión U_f a todos los estados, de modo que:

$$U_f|\psi_t\rangle = |\psi_{t'}\rangle,$$

donde, como vimos anteriormente, si $|\psi_t\rangle$ es el estado $|w\rangle$, este se puede representar geométricamente como la reflexión negativa $-|w\rangle$. Por otro lado, si son estados no marcados $|x\rangle$, la función no tiene ningún efecto sobre ellos.

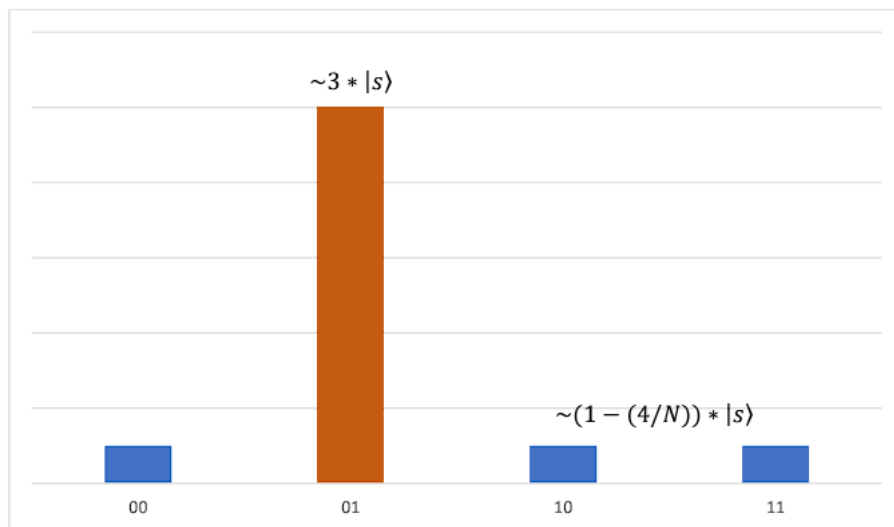
Figura 2: Amplitudes transformación U_f paso 2

Y por último, en el tercer paso, aplicamos una reflexión en el estado $|s\rangle$, la cual se escribe como:

$$U_s = 2|s\rangle\langle s| - I,$$

donde I es la matriz identidad. Esta transformación modifica todos los estados a $U_s|\psi_{t'}\rangle$ y completa la transformación:

$$|\psi_{t+1}\rangle = U_s U_f |\psi_{t'}\rangle.$$

Figura 3: Amplitudes transformación U_s paso 3

Matemáticamente, U_s se puede representar como la multiplicación de las matrices:

$$U_s = 2 \begin{bmatrix} 1/\sqrt{N} \\ \vdots \\ 1/\sqrt{N} \end{bmatrix} \begin{bmatrix} 1/\sqrt{N} & \dots & 1/\sqrt{N} \end{bmatrix} - I = \begin{bmatrix} \frac{2}{N} - 1 & \frac{2}{N} & \dots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} - 1 & \dots & \frac{2}{N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{2}{N} & \frac{2}{N} & \dots & \frac{2}{N} - 1 \end{bmatrix}$$

Por lo tanto, resumiendo un poco por medio de sumatorias.

$$\begin{aligned} U_s |\psi_t\rangle &= (2|s\rangle\langle s|-1)|\psi_t\rangle \\ U_s |\psi_t\rangle &= \left(2 \left(\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \right) \left(\frac{1}{\sqrt{N}} \sum_{x'=0}^{N-1} \langle x'| \right) - 1 \right) \sum_{x''=0}^{N-1} \alpha_{x''} |x''\rangle \\ U_s |\psi_t\rangle &= \left(\frac{2}{N} \sum_{x=0, x'=0, x''=0}^{N-1} \alpha_{x''} |x\rangle \langle x'| x''\rangle - \sum_{x=0}^{N-1} \alpha_x |x\rangle \right) \\ U_s |\psi_t\rangle &= \left(\frac{2}{N} \sum_{x''=0}^{N-1} \alpha_{x''} \sum_{x=0}^{N-1} |x\rangle - \sum_{x=0}^{N-1} \alpha_x |x\rangle \right) \\ U_s |\psi_t\rangle &= \sum_{x=0}^{N-1} \left(2 \sum_{x''=0}^{N-1} \frac{\alpha_{x''}}{N} - \alpha_x \right) |x\rangle \\ U_s |\psi_t\rangle &= \sum_{x=0}^{N-1} (2A - \alpha_x) |x\rangle \end{aligned}$$

Donde A es el promedio de cada α_x , dado por:

$$A = \frac{\sum_{x''=0}^{N-1} \alpha_{x''}}{N}.$$

Después de algunas reducciones por las sumatorias y el promedio, llegamos al estado:

$$|\psi_{t+1}\rangle = U_s U_f |\psi_t\rangle.$$

$$|\psi_{t+1}\rangle = \left[\frac{2^{n+1} + 2^n - 4}{2^n \sqrt{2^n}} \right] |x\rangle \quad \text{Para } x = \omega$$

$$|\psi_{t+1}\rangle = \left[\frac{2^{n+1} - 2^n - 4}{2^n \sqrt{2^n}} \right] |x\rangle \quad \text{Para } x \neq \omega$$

Al implementar el algoritmo en un computador cuántico, este repite los 3 pasos y se obtiene una respuesta más acertada. Con estas dos reflexiones U_s y U_f , obtenemos una rotación y una ampliación de la amplitud. Geométricamente, es una ampliación sobre el promedio de las amplitudes, como lo vemos en el paso 2, donde en la gráfica el valor de $|s\rangle$ es menor a las amplitudes de los estados no marcados, ya que el estado marcado (Estado 2) es negativo. Y por último en el paso 3 al volver hacer una reflexión este valor negativo es convertido en positivo.

Implementación del algoritmo de Grover en Qiskit

In [2]:

```
# Importamos Las librerías necesarias
```

```
from qiskit import QuantumCircuit, Aer, execute
from qiskit.visualization import plot_histogram
```

```
# Implementamos La puerta de Grover
```

```
Grover = QuantumCircuit(3,3)
```

```
Grover.h(range(3))
```

```
Grover.x(range(3))
```

```
#CORRECCIÓN !!!
```

```
# Si el ccz Lo implementamos de esta manera estaremos cambiando algunas amplitudes que no de  
# La solución está en recordar que Z = HXH
```

```
Grover.h(2)
```

```
Grover.mct([0,1],2,0) # esto es el cccx, el primer parámetro indica Los qubits de control, e
```

```
Grover.h(2) # el qubit objetivo. Finalmente pond un cero, no deis importancia a
```

```
#Grover.x(2)
```

```
#Grover.z(2)
```

```
#Grover.x(2)
```

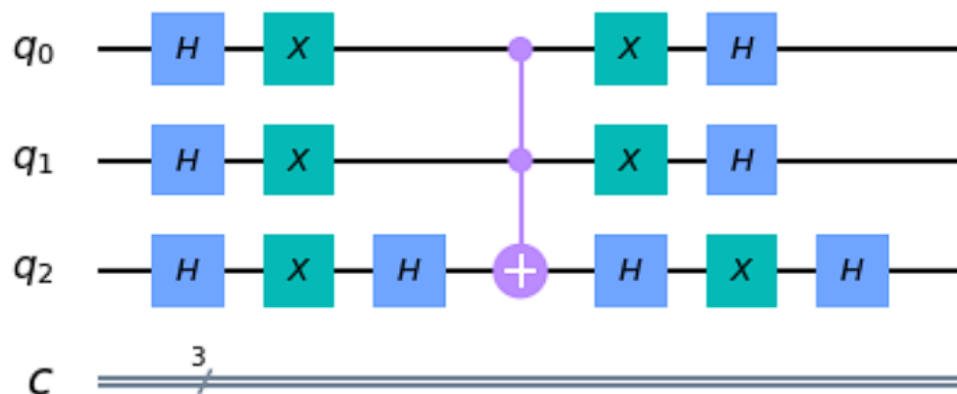
```
#Grover.mct(range(2),2,0)
```

```
Grover.x(range(3))
```

```
Grover.h(range(3))
```

```
Grover.draw(output = "mpl")
```

Out[5]:



In [6]:

```
# Esta es la función que podeis cambiar para buscar nuevos elementos
```

```
Detector = QuantumCircuit(3,3)
```

```
Detector.cz(1,0)
```

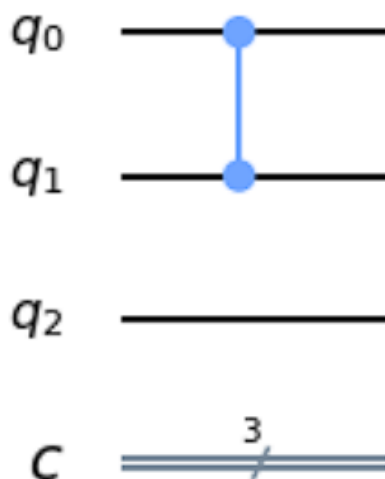
```
Detector.draw(output = "mpl")
```

```
Detector = QuantumCircuit(3,3)
```

```
Detector.cz(1,0)
```

```
Detector.draw(output = "mpl")
```

Out[6]:

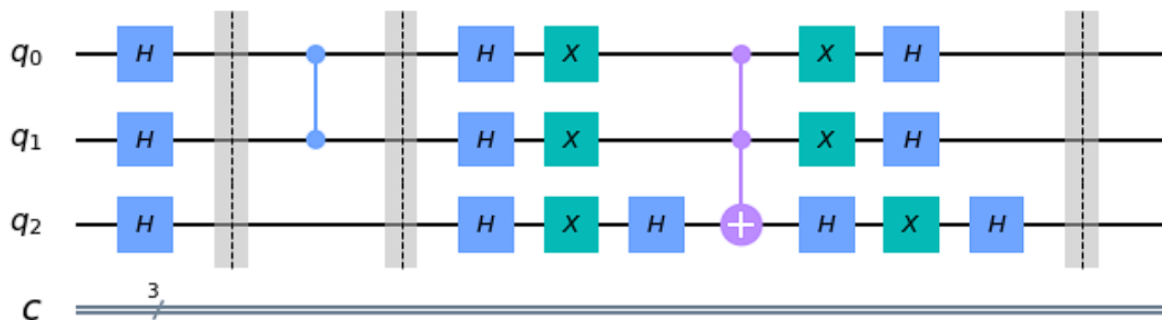


```

circ = QuantumCircuit(3,3)
circ.h(range(3))
circ.barrier(range(3))
circ = circ + Detector
circ.barrier(range(3))
circ = circ + Grover
circ.barrier(range(3))
#circ.measure(range(3), range(3))
circ.draw(output = "mpl")

```

Out[7]:



In [10]:

```

backend = Aer.get_backend("qasm_simulator")
job = execute(circ, backend, shots = 1000)
result = job.result()
counts = result.get_counts()
plot_histogram(counts)

```

Out[10]:

