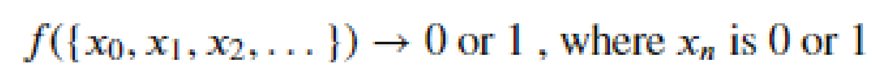
Quantum algorithms

- Deutsch-Jozsa algorithm
  - We are given a hidden Boolean function f , which takes as input a string of bits, and returns either 0 or 1 , that is:



$$f(\{x_0, x_1, x_2, \dots\}) \to 0 \text{ or } 1 \text{ , where } x_n \text{ is } 0 \text{ or } 1$$

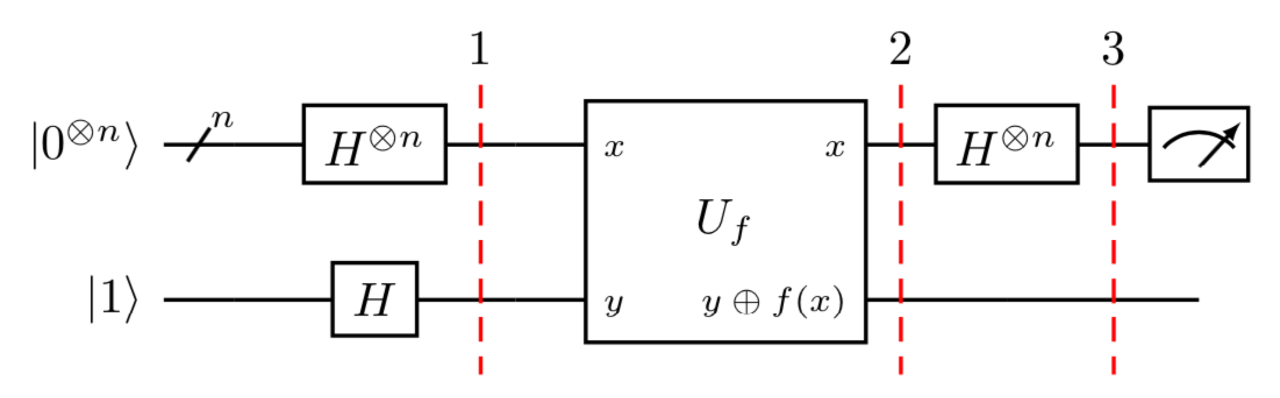
  - The property of the given Boolean function is that it is guaranteed to either be balanced (returns 1 for half of the input domain and 0 for the other half) or constant (0 on all inputs or 1 on all inputs).
  - Our task is to determine whether the given function is balanced or constant.

Quantum algorithms

- Deutsch-Jozsa algorithm
  - ➢ For classical solution, we need to ask the oracle at least twice, but if we get twice the same output, we need to ask again. At most to query is (*N*/2)+1, where *N* is number of state.
  - ➢ For quantum solution, need only one query. If the output is **the zero bit string**, we know that the oracle is **constant**. If it is **any other bit string**, we know that it is **balanced**.
  - ➢ We have the function f implemented as a quantum oracle, which maps the state $|x\rangle|y\rangle$ to $|x\rangle|y \oplus f(x)\rangle$, where $\oplus$ is addition modulo 2.

Quantum algorithms

- Deutsch-Jozsa algorithm



  - ➢ The initial state of which can be expressed:

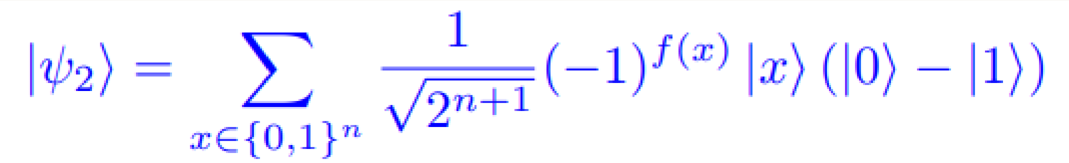
$$|\psi_0\rangle = |0\rangle^{\otimes n} |1\rangle$$

  - ➢ which is then put into superposition, which can conveniently be expressed:

$$|\psi_1\rangle = \sum_{x \in \{0,1\}^n} \frac{1}{\sqrt{2^{n+1}}} |x\rangle (|0\rangle - |1\rangle)$$

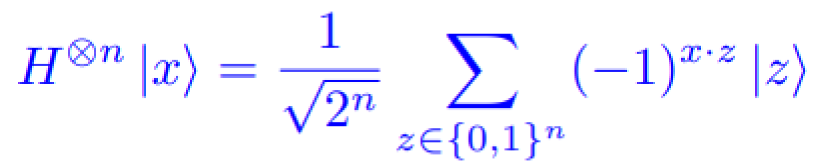
Quantum algorithms

- Deutsch-Jozsa algorithm
  - Apply the quantum oracle $|x\rangle|y\rangle$ to $|x\rangle|y\oplus f(x)\rangle$:



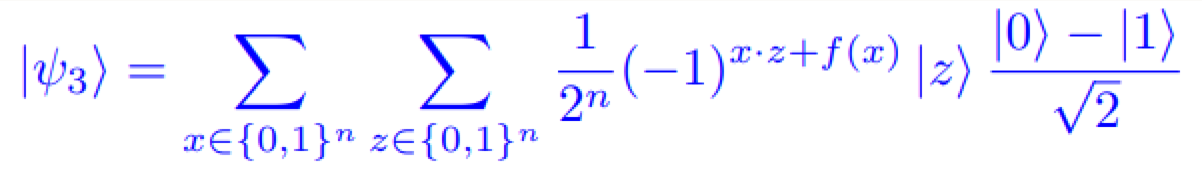
$$|\psi_2\rangle = \sum_{x\in\{0,1\}^n} \frac{1}{\sqrt{2^{n+1}}}(-1)^{f(x)}\,|x\rangle\,(|0\rangle - |1\rangle)$$

  - We now address the interference H on the first n wires, for which we use the expression:



$$H^{\otimes n}\,|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} (-1)^{x\cdot z}\,|z\rangle$$

  - which allows us to express:

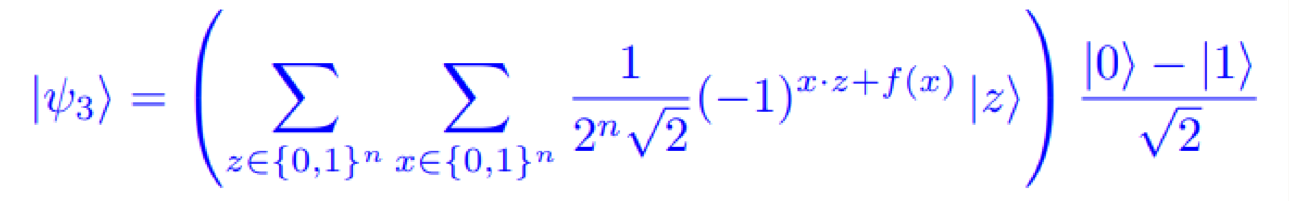


$$|\psi_3\rangle = \sum_{x\in\{0,1\}^n} \sum_{z\in\{0,1\}^n} \frac{1}{2^n}(-1)^{x\cdot z+f(x)}\,|z\rangle\,\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

  - where $x.z = x_0z_0 \oplus x_1z_1 \oplus \ldots \oplus x_{n-1}z_{n-1}$ is the sum of the bitwise product.

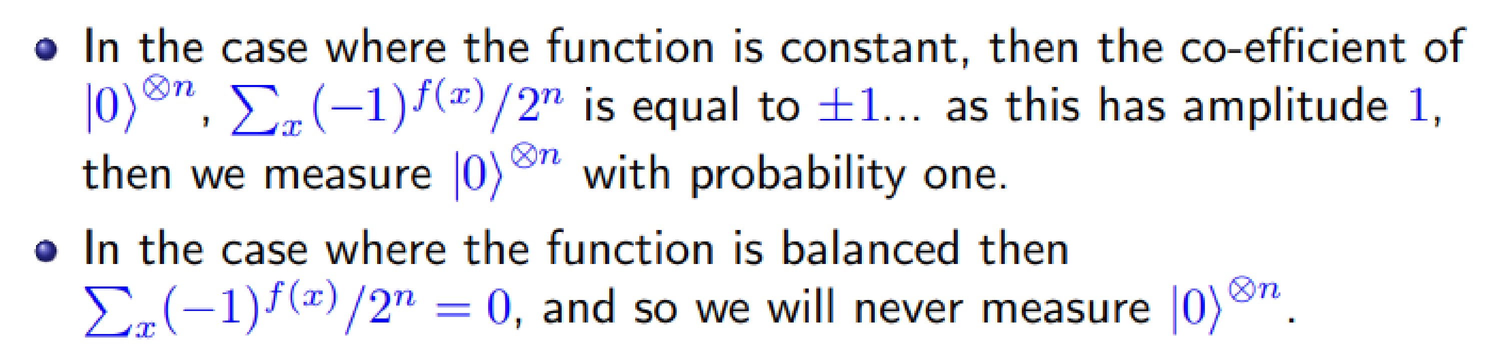
Quantum algorithms

- Deutsch-Jozsa algorithm
  - We can now determine whether the function is constant or balanced by measuring the first *n* qubits of the final state.



$$|\psi_3\rangle = \left(\sum_{z \in \{0,1\}^n} \sum_{x \in \{0,1\}^n} \frac{1}{2^n \sqrt{2}} (-1)^{x \cdot z + f(x)} |z\rangle \right) \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

  - Specifically, we consider the probability of measuring zero on every qubit, which corresponds to the term in the superposition where $|x\rangle$ is $|0\rangle^{\otimes n}$



- In the case where the function is constant, then the co-efficient of $|0\rangle^{\otimes n}$, $\sum_x (-1)^{f(x)}/2^n$ is equal to ± 1... as this has amplitude 1, then we measure $|0\rangle^{\otimes n}$ with probability one.
- In the case where the function is balanced then $\sum_x (-1)^{f(x)}/2^n = 0$, and so we will never measure $|0\rangle^{\otimes n}$.

  - So it follows that measuring the first *n* qubits allows us to determine with certainty whether the function is **constant (measure all zeros)** or **balanced (measure at least one 1)**.

Quantum algorithms

- **Deutsch-Jozsa algorithm**
  - We can encode any mathematical function as a unitary matrix.
  - Deutsch's algorithm was the first algorithm that demonstrated a quantum advantage: specifically, a reduction in query complexity compared to the classical case.
  - The Deutsch-Jozsa algorithm generalises Deutsch's algorithm and reveals the possibility of exponential speed-ups using quantum computers.

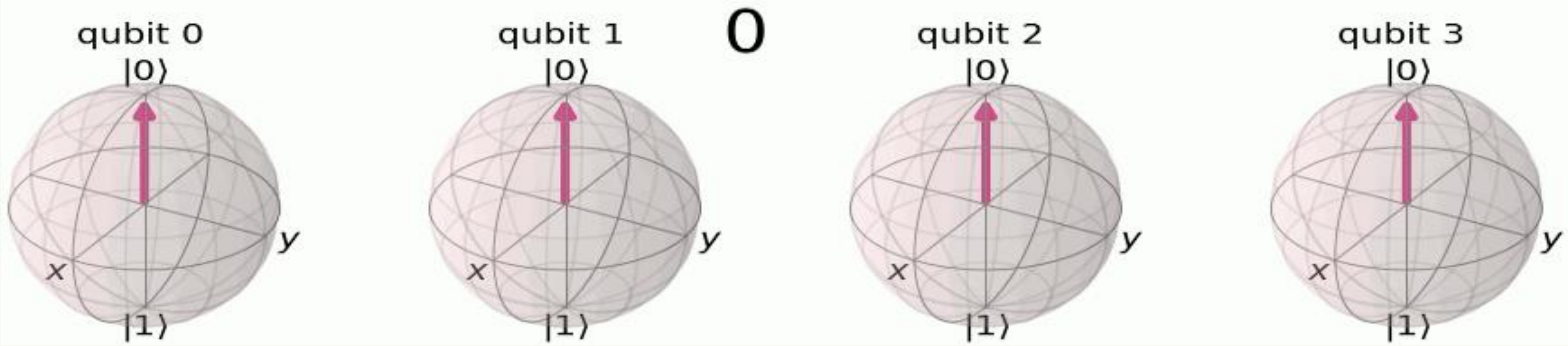
Quantum algorithms

- **Quantum Fourier Transform (QFT)**
  - ➢ The QFT is the quantum implementation of the discrete Fourier transform over the amplitudes of a wavefunction.
  - ➢ The QFT simply transforms a qubit from its computational basis of $|0\rangle$ and $|1\rangle$ to the state in Fourier basis $|+\rangle$ and $|-\rangle$.

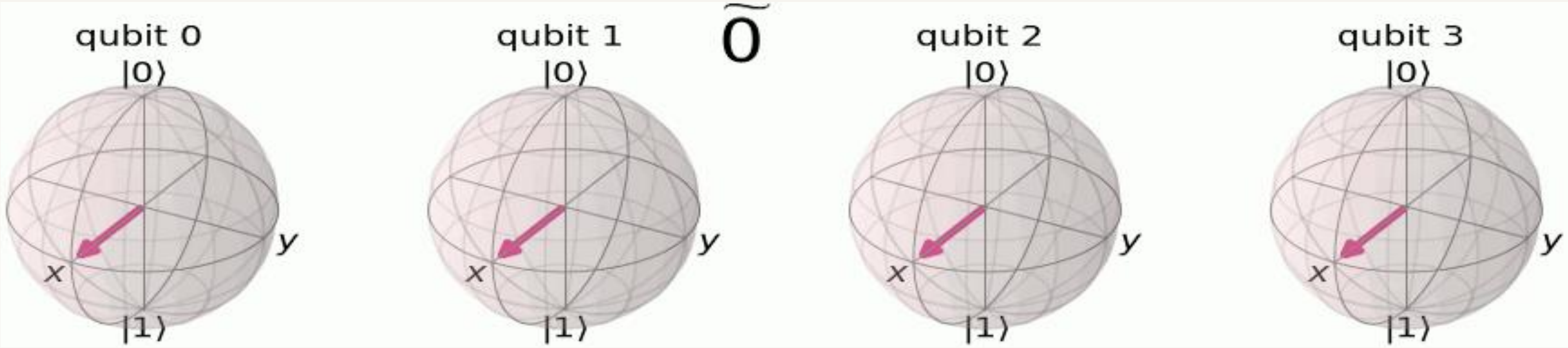
Quantum algorithms

*Try it out at AssignmentII and upload files “**quantum_fourier_transform.ipynb**” into IBM Quantum Lab.*

- **Quantum Fourier Transform (QFT)**
  - ➢ Computational basis:



  - ➢ Fourier basis:

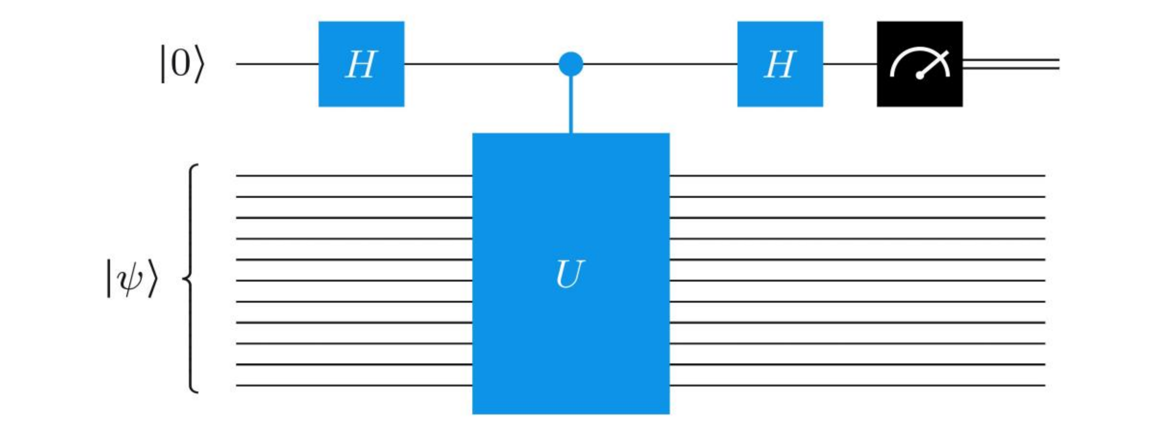


Quantum algorithms

- **Quantum Phase Estimation (QPE)**
  - ➢ QPE aims to estimate the phase θ associated with an eigenvalue $e^{2\pi i\theta}$ of a unitary operator U.
  - ➢ The quantum phase estimation algorithm uses phase kickback to write the phase of U, in the Fourier basis, to the t qubits in the counting register.

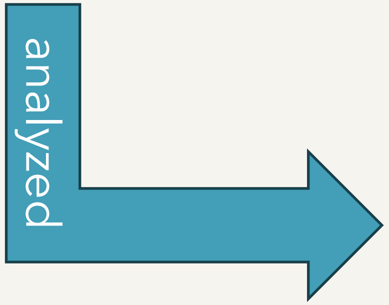
Quantum algorithms

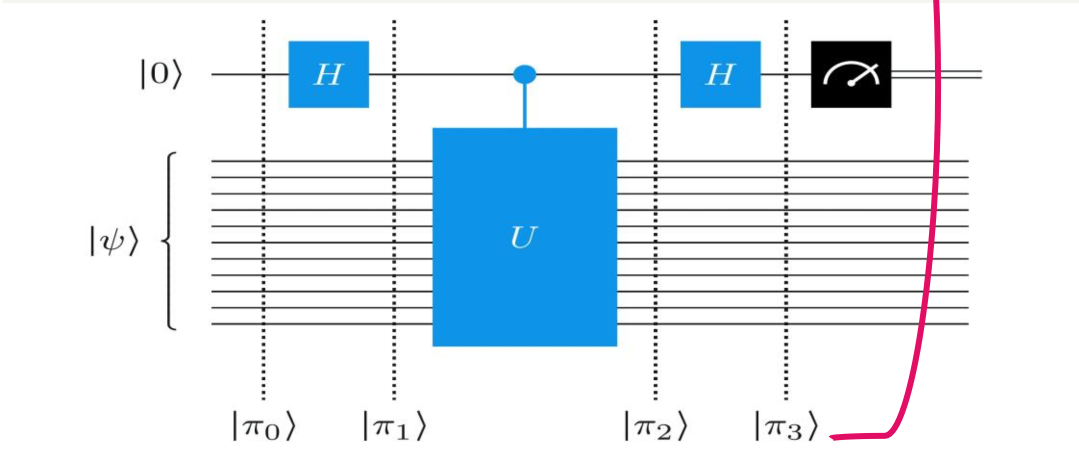
- **Quantum Phase Estimation (QPE): Single qubit**



$$p_0 = \left|\frac{1 + e^{2\pi i\theta}}{2}\right|^2 = \cos^2(\pi\theta)$$

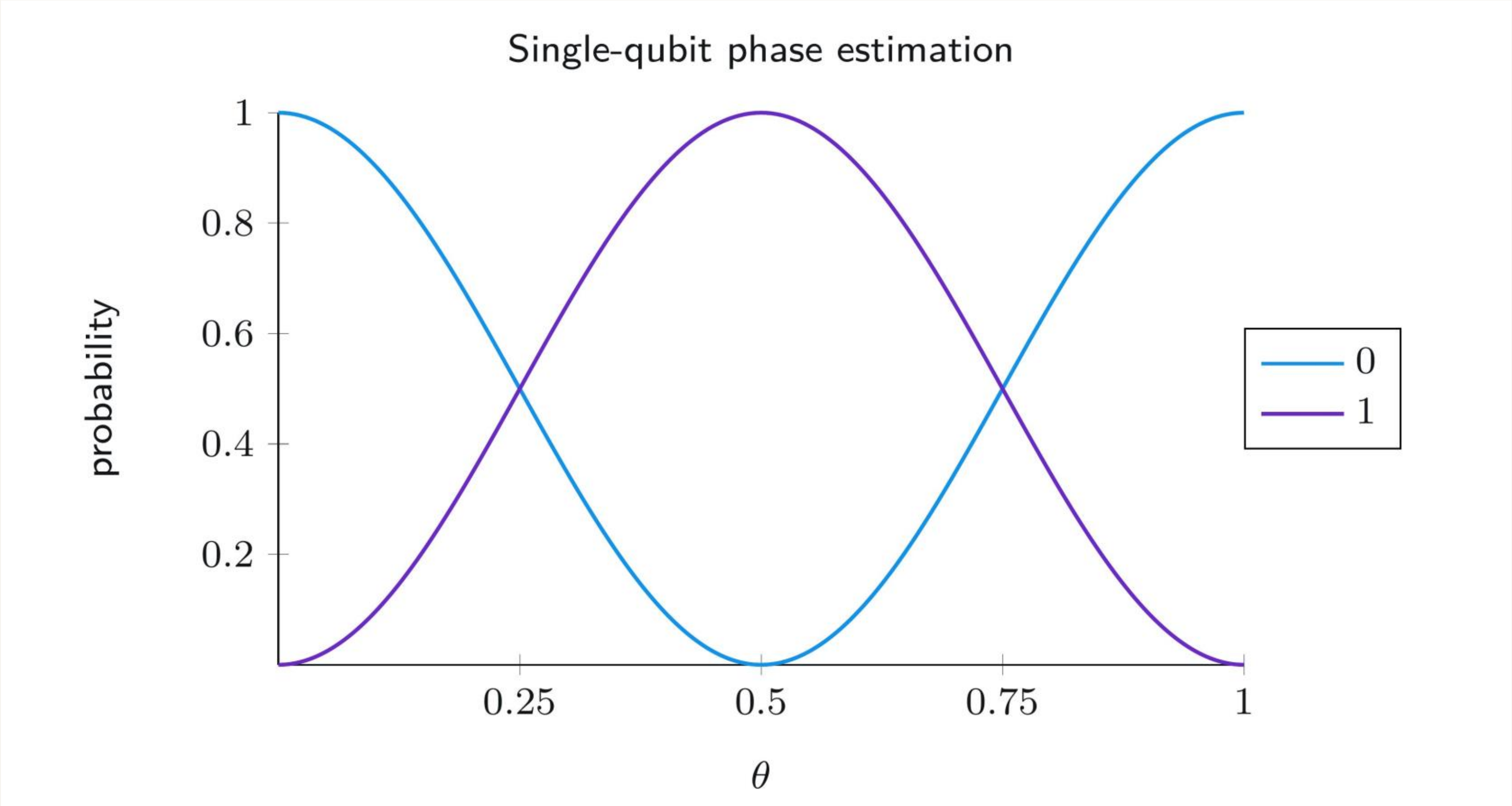
$$p_1 = \left|\frac{1 - e^{2\pi i\theta}}{2}\right|^2 = \sin^2(\pi\theta).$$





Quantum algorithms

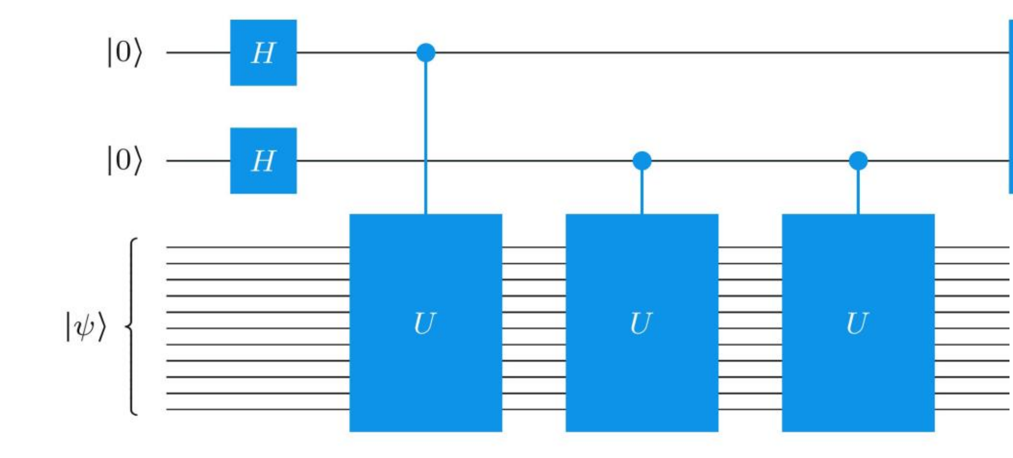
- **Quantum Phase Estimation (QPE)**

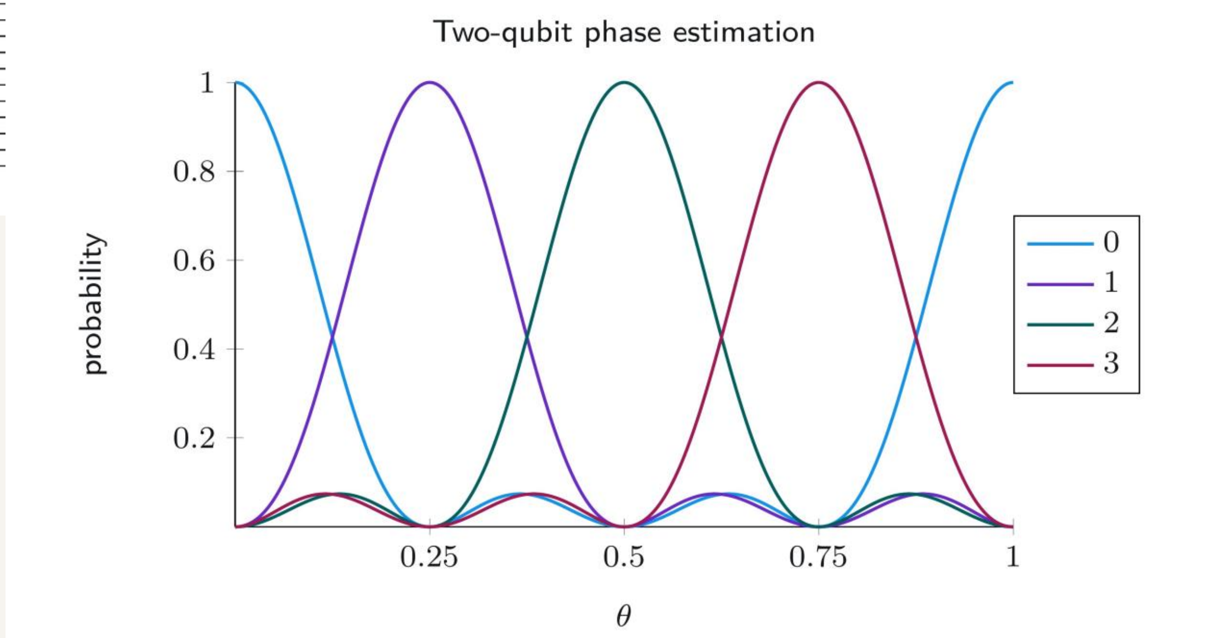


Quantum algorithms

- **Quantum Phase Estimation (QPE): Two qubits**

Try it out at *AssignmentII* *and upload files "****phase_estimation.ipynb****" into IBM Quantum Lab.*





Quantum algorithms

- **Shor's algorithm**
  - Let N be the integer we want to factor. Let's assume the example is number 35.
  - Pick a random integer from 2 to N-1. Let's call this number a. Let's assume a is 4.
  - Find the greatest common divisor (GCD) between a and N. If you get a value that is not 1, it means that the GCD obtained is the answer. It's finished. You don't have to do anything further. But if it is equal to 1, see the next step.
  - Find the value of the function $f(x) = a^x \; mod \; n$.
  - From the example N=35, a=4, the table between the values of x and f(x) will be obtained as follows.

X	0	1	2	3	4	5	6	7	8	9
f(x)	1	4	16	29	11	9	1	4	16	29

  - We have to check that $a^{r/2} = -1 \; (mod \; n)$. If so, we have to random new "a".
  - Then we find the GCD between $(a^{r/2} + 1, N)$ and $(a^{r/2} - 1, N)$. If we get 1 and N, go back to random new "a" again.

Quantum algorithms

- **Shor's algorithm**
  - A reduction of the factoring problem to the problem of order-finding, which can be done on a classical computer.

  - A quantum algorithm to solve the order-finding problem.

Quantum algorithms

- **Shor's algorithm**

➢ Classical part

1. Pick a pseudo-random number a < N
2. Compute gcd(a, N). This may be done using the Euclidean algorithm.
3. If gcd(a, N) ≠ 1, then there is a nontrivial factor of N, so we are done.
4. Otherwise, use the period-finding subroutine (below) to find r, the period of the following function:

   $f(x) = a^x \; mod \; N$, i.e. the smallest integer *r* for which $f(x + r) = f(x)$.
5. If r is odd, go back to step 1.
6. If $a^{r/2} = -1 \; (mod \; n)$ go back to step 1.
7. The factors of N are gcd($a^{r/2} \pm 1, N$). We are done.

Quantum algorithms

- **Shor's algorithm**

➢ Quantum part: Period-finding subroutine

1. Start with a pair of input and output qubit registers with $log_2 n$ qubits each, and initialize them to

   $N^{-1/2}\sum_x|x\rangle|0\rangle$, where x runs from 0 to N-1

2. Construct f(x) as a quantum function and apply it to the above state, to obtain

   $N^{-1/2}\sum_x|x\rangle|f(x)\rangle$

3. Apply the quantum Fourier transform on the input register. The quantum Fourier transform on N points is defined by:

   $U_{QFT}|x\rangle = N^{-1/2}\sum_y e^{2\pi ixy/N}|y\rangle$

   This leave us in the following state:

   $N^{-1}\sum_x\sum_y e^{2\pi ixy/N}|y\rangle\,|f(x)\rangle$

4. Perform a measurement. We obtain some outcome y in the input register and $f(x_0)$ in the output register. Since f is periodic, the probability to measure some y is given by:

   $N^{-1}\left|\sum_x : f(x) = f(x_0) e^{2\pi ixy/N}\right|^2 = N^{-1}\left|\sum_b e^{2\pi i(x_0+r_b)y/N}\right|^2$

   Analysis now shows that this probability is higher, the closer y/N is to an integer.

Quantum algorithms

- **Shor's algorithm**

➢ Quantum part: Period-finding subroutine

5. Turn y/N into an irreducible fraction, and extract the denominator r', which is a candidate for r.
6. Check if $f(x) = f(x + r')$. If so, we are done.
7. Otherwise, obtain more candidates for r by using values near y, or multiples of r'. If any candidate works, we are done.
8. Otherwise, go back to step 1 of the subroutine.

*Try it out at AssignmentII and upload files "**Shor's algorithm.ipynb**"*
into IBM Quantum Lab.

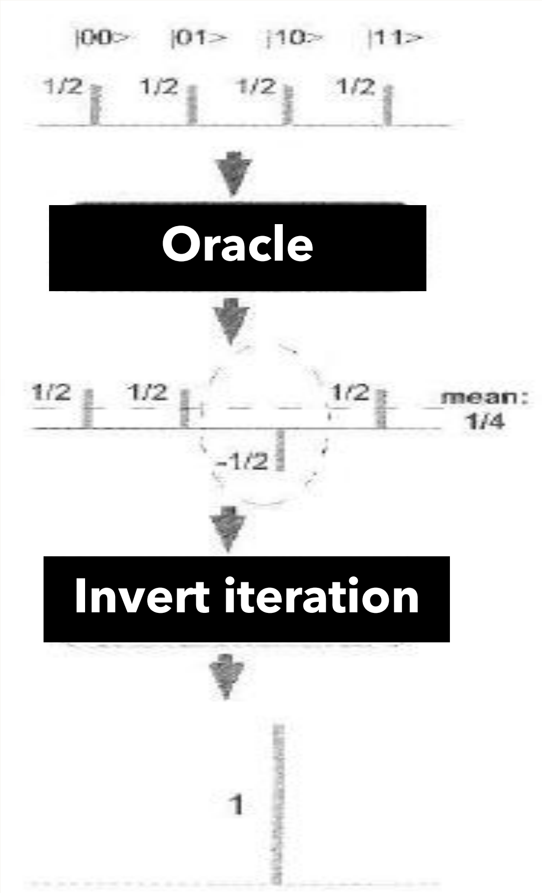
Quantum algorithms

- **Grover's algorithm**
  - It can be used to solve unstructured search problems in roughly $\sqrt{N}$ steps, where *N* is the amount of data.
  - This algorithm can speed up an unstructured search problem quadratically using the amplitude amplification trick.

4	**6**	**8**		**W**		**N=2^n**

Quantum algorithms

- Operation of searching data by Grover's algorithm for 2 qubits:



$$\frac{1}{2}(|00\rangle+|01\rangle+|10\rangle+|11\rangle)$$

$$\frac{1}{2}(|00\rangle+|01\rangle-|10\rangle+|11\rangle)$$

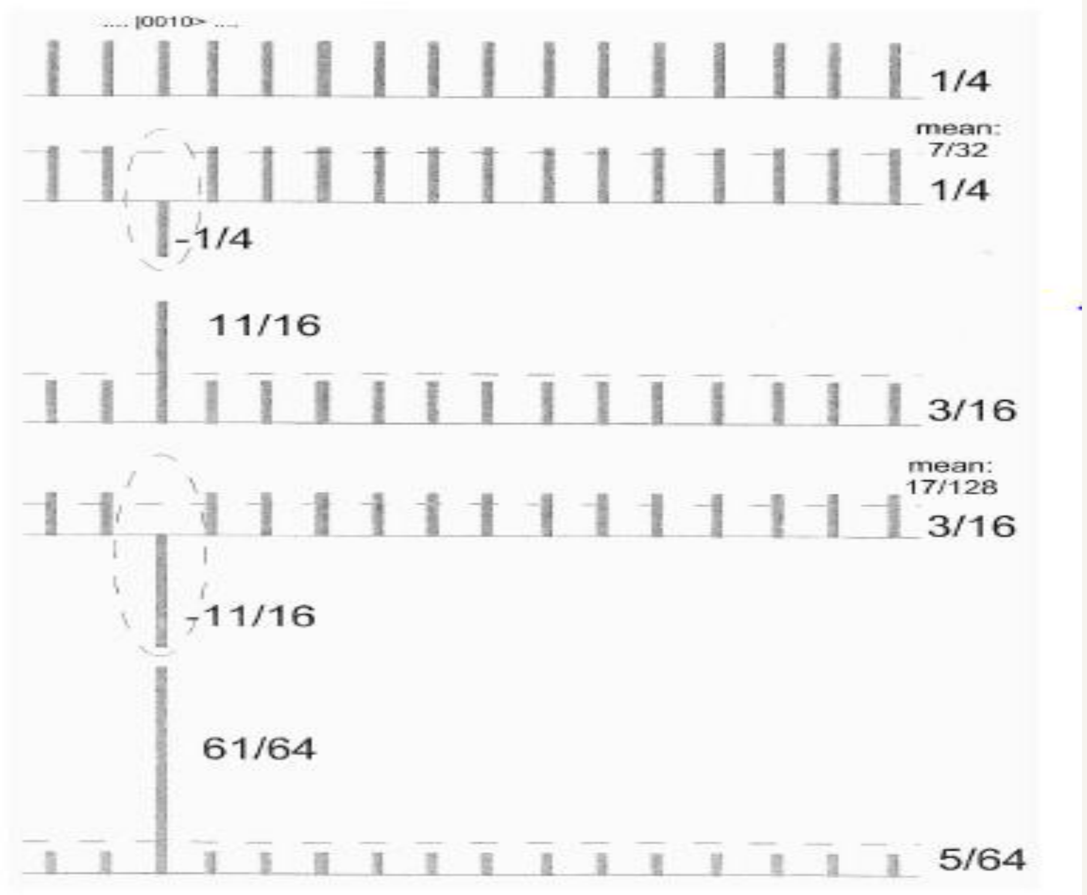
$$m=\frac{\left(\frac{1}{2}+\frac{1}{2}-\frac{1}{2}+\frac{1}{2}\right)}{4}=\frac{1}{4}$$

$$l_i|00\rangle,|01\rangle,|11\rangle=\frac{1}{4}-\left(\frac{1}{2}-\frac{1}{4}\right)=0$$

$$l_i|10\rangle\ =\frac{1}{4}-\left(-\frac{1}{2}-\frac{1}{4}\right)=1$$

Quantum algorithms

- Operation of searching data by Grover's algorithm for 4 qubits:



Grover iterations = $\frac{\pi}{4}\ x\ \sqrt{\frac{N}{t}}$ times,

N is the number of data (states) and
t is the number of target solutions.

Try it out at AssignmentII *and upload files "**Grover's algorithm.ipynb**" into IBM Quantum Lab.*

Quantum algorithms

- Grover's algorithm
  - ➢ The example of Grover's algorithm for 3 qubits with two marked states $|101\rangle$ and $|110\rangle$.

Grover iterations ~ $\frac{\pi}{4}\sqrt{\frac{N}{t}}$

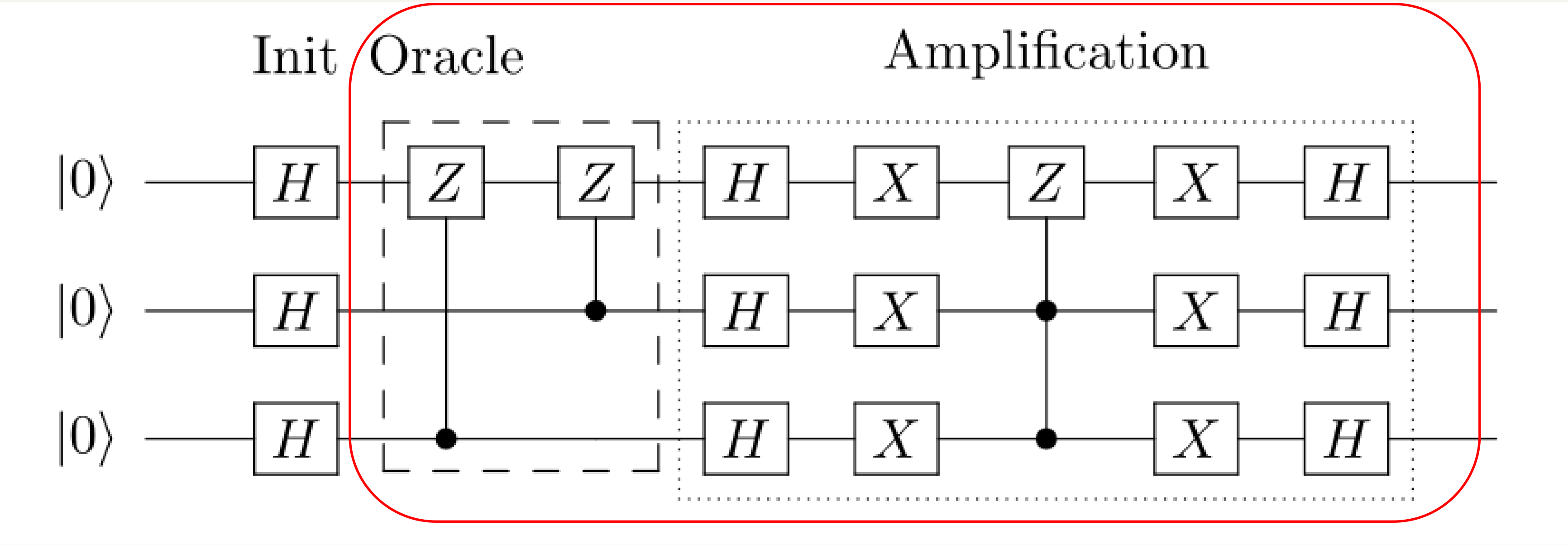


Photo courtesy of https://qiskit.org/textbook/ch-algorithms/grover.html

Quantum algorithms

- The implemented stages of the Grover's search algorithm:
  - Initialization: In the first stage of the algorithm all qubits are set to be in superposition by applying the Hadamard gate to each qubit. After this operation the amplitude of each state is 1/sqrt(n).
  - Oracle: The oracle function performs a phase flip on the marked state. If the marked state is |0110〉 , the phase flip inverts the amplitude α0110 of the state.
  - Amplification: The amplification stage performs an inversion of the average of the amplitudes.
  - Measurement: The qubits are measured in finally.

Grover iterations ~ $\frac{\pi}{4}\sqrt{\frac{N}{t}}$

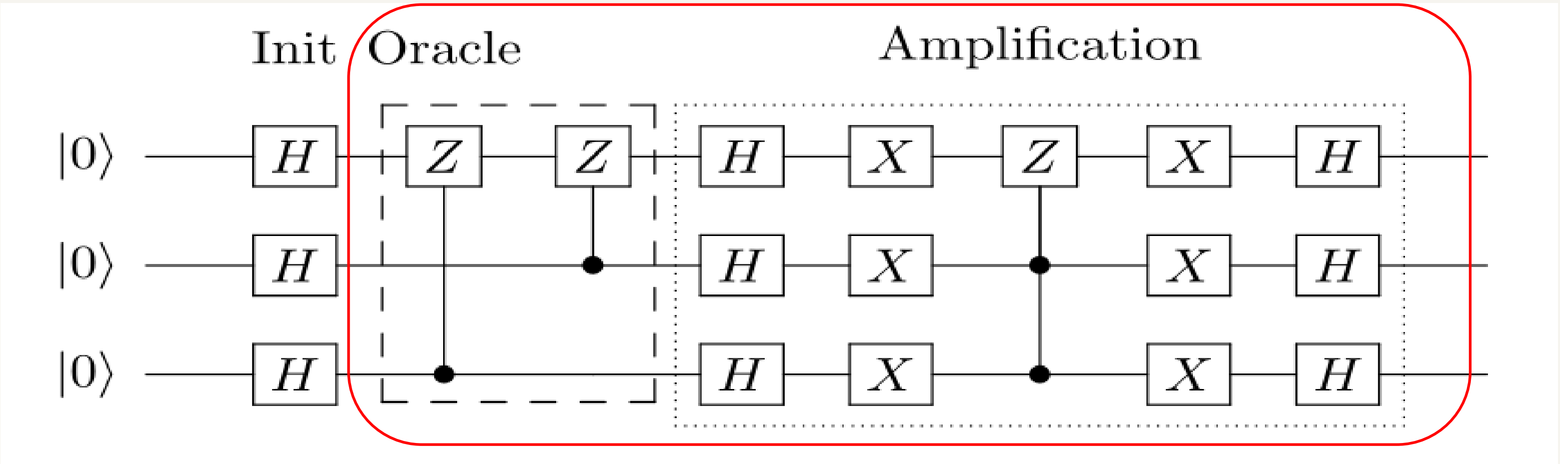
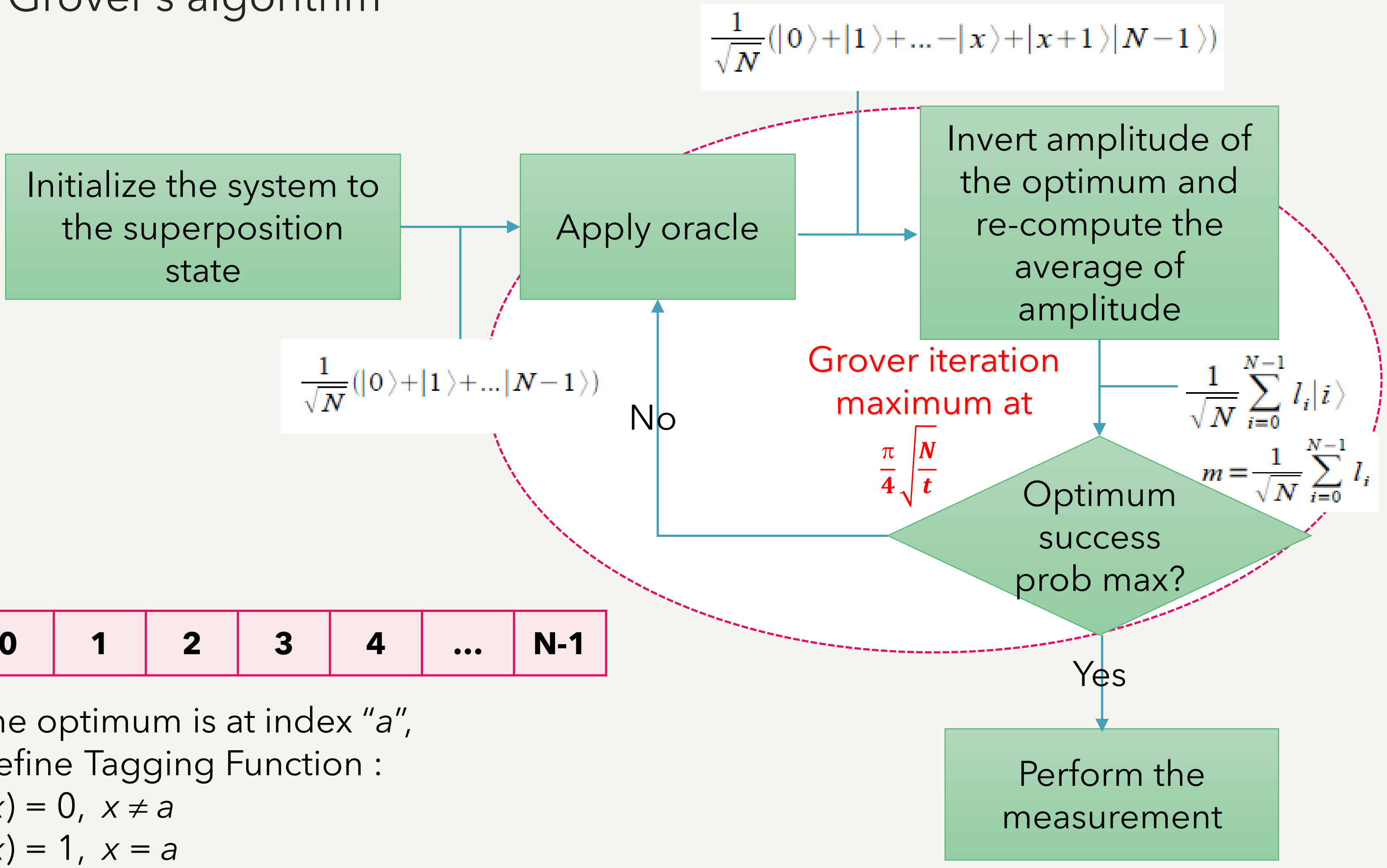


Photo courtesy of https://qiskit.org/textbook/ch-algorithms/grover.html

Quantum algorithms

- Grover's algorithm



$\frac{1}{\sqrt{N}}(|0\rangle+|1\rangle+\ldots-|x\rangle+|x+1\rangle|N-1\rangle)$
Initialize the system to the superposition state
Apply oracle
Invert amplitude of the optimum and re-compute the average of amplitude
$\frac{1}{\sqrt{N}}(|0\rangle+|1\rangle+\ldots|N-1\rangle)$
No
Grover iteration maximum at
$\frac{\pi}{4}\sqrt{\frac{N}{t}}$
$\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1} l_i|i\rangle$
$m=\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1} l_i$
Optimum success prob max?
Yes
Perform the measurement

New amplitude:

$$l_{i\_new} = m - (l_i - m)$$
$$= 2m - l_i$$

m is new average of amplitude.

0	1	2	3	4	…	N-1

The optimum is at index "*a*",
Define Tagging Function :
f(x) = 0, $x \neq a$
f(x) = 1, $x = a$

Assignment II: quantum algorithms

- Required:
  - Go to https://quantum-computing.ibm.com/
  - Download source codes at Assignment and upload files "**Lab-4.ipynb**" into IBM Quantum Lab.
- Assignment:
  - Lab-4: Oracles and the Deutsch-Jozsa algorithm by IBM Quantum.