The supremacy of quantum algorithms
Lesson 4 – The Grover Algorithm

Imagine you want to crack a safe with a combination lock that has four digits ranging from 0 to 9. You have no clue which numbers might be correct. If you're very lucky, you might hit the right combination on your first try. How many attempts would you need if things go badly? How many attempts would you need on average? Remember the search in unstructured data in lesson 1.

The search problem

There is a list of N numbers, in which exactly one of the numbers is assigned to 1 (this is the right number) and all other numbers are assigned to 0. The task is to find the position of the number which is assigned to 1 with the least possible number of questions.

The mathematical description of this problem can be done in terms of a function f with the following definition. There exists exactly one r from 1 to N with:

$$f(x) = \begin{cases} 1 & if\ x = r \\ 0 & for\ all\ other\ values\ of\ x \end{cases}$$

Classical solution

Exercise 1: Classical solution

Assuming you don't know what r is, determine the average number of guesses of x you will input into the function f to find r. What is your strategy for doing this?

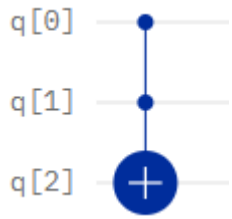
Exercise 2: Binary representation

We will use the binary representation of the numbers from 1 to N. So $N = 2^n$ and r is represented as a string of 0s and 1s of length n.

Find the binary representation of r = 5 in the case N = 16.

Implementing the search function using quantum gates

In order to implement the search function, we need to introduce a new gate, known as the **Toffoli gate**. This is very similar to a CNOT gate, but acts on multiple input qubits. For example:



This Toffoli gate will perform a NOT operation on the output qubit q[2] only when both input qubits, q[0] and q[1], are in state $|1\rangle$. With the output qubit initially set to state $|0\rangle$, this structure has the same effect as the equivalent to the AND operation in classical computing.

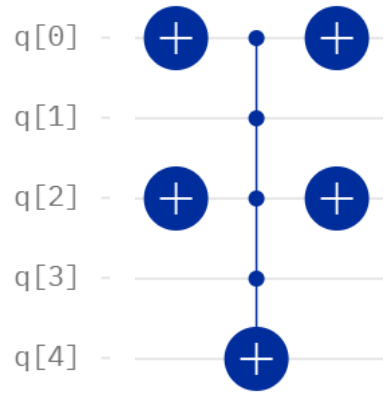
The following table shows the final state of the output qubit, q[2], for all combinations of the two input qubits:

q[0]	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
q[1]	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
q[2] (final)	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$

To build the hidden function for the Grover search, we use a combination of an n-input Toffoli gate and single qubit NOT gates.

For our implementation of the Grover Algorithm, we will use a 4-qubit implementation where we are searching for the hidden string 0101. To build the circuit for this, we need a 4-qubit Toffoli gate and NOT gates for the qubits that correspond to 0s in the hidden string. The NOT gates flip these two inputs to $|1\rangle$ so that the Toffoli gate flips the state of the output qubit to $|1\rangle$ only when all of the input states match the hidden string. After the Toffoli gate the corresponding states have to flipped back by NOT gates.

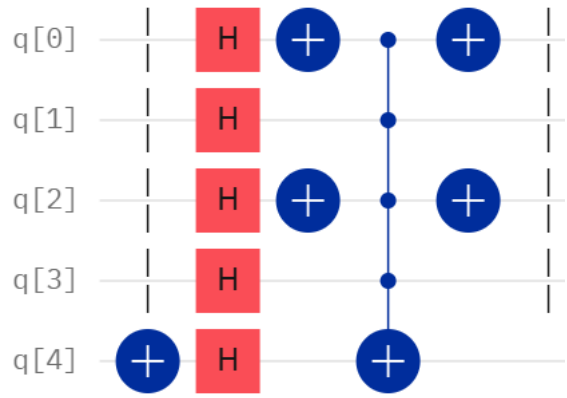
Gate structure for hidden string 0101



The difference between Grover and Bernstein-Vazirani

In the previous lesson, we saw how the Bernstein-Vazirani hidden function could be found in a single step. However, the Bernstein-Vazirani function had the property that it could be constructed using only CNOT gates, and this was the reason why the application of Hadamards led immediately to the discovery of the hidden function.

The objective of Grover is to solve the more realistic problem of identifying whether a particular data item exists within an unstructured set of data. The requirement to use Toffoli gates to do this means that the Bernstein-Vazirani approach does not give us the desired result. However, as in both Deutsch and Bernstein-Vazirani, we begin by putting all input qubits into state $|0\rangle$, the output qubit into state $|1\rangle$, and applying Hadamards to all qubits:



The following table shows the effect on the input qubits happens of this structure:

	States															
	0 |0000>	1 |0001>	2 |0010>	3 |0011>	4 |0100>	5 |0101>	6 |0110>	7 |0111>	8 |1000>	9 |1001>	10 |1010>	11 |1011>	12 |1100>	13 |1101>	14 |1110>	15 |1111>
	Amplitudes															
Hadamard	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Function	0.25	0.25	0.25	0.25	0.25	-0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25

The application of the Hadamards creates a superposition of 16 different states (all the binary combinations of 0s and 1s from 0 to 15). Note that the probability of being in any of a measurement resulting in any particular state is 1/16 and the initial amplitude of each is 1/4. This is the same starting point as in all of our previous algorithms.

However, the application of the hidden function then has a curious effect. All of the amplitudes of the incorrect guesses remain the same, but the coefficient of the correct combination is flipped to have a negative sign.

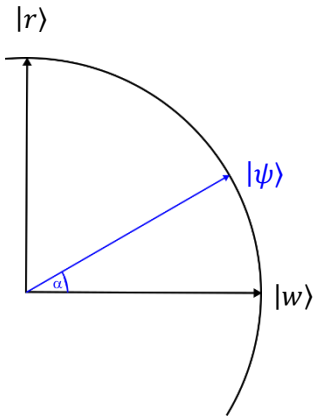
It is important to understand that this alone will not allow us to find the hidden string. The probability of being in each state is the square of the amplitude, so this state still has the same probability as all others and measurement at this stage would not give us any better chance of finding the hidden string.

However, if we apply further operations to this superposition state, we are able to amplify the magnitude of the state of the hidden string (whilst shrinking all of the others). The following part shows how Grover's Algorithm does this.

Grover's Algorithm: Quantum Solution

Grover's algorithm tackles our problem using only $\sqrt{N}$ number of questions, thanks to the power of quantum computing. The algorithm is implemented in the following steps, which we will understand in a geometric picture:

Step I: Prepare n qubits in state $|0\rangle$ and apply a Hadamard gate to all of them. This gives a superposition of all possible states on the n qubits.
This state $|\psi\rangle$ can be described in a composition of the desired (right) state $|r\rangle$ and the (normalized) sum of all other (wrong) states $|w\rangle$, which are perpendicular to $|r\rangle$.

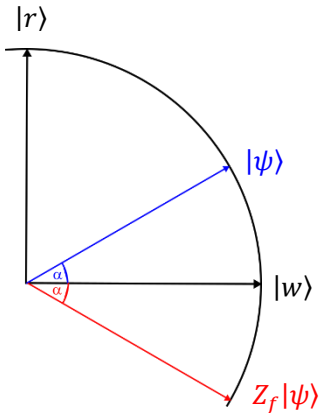


Example:
In the case of N = 4 we will get $|\psi\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$.

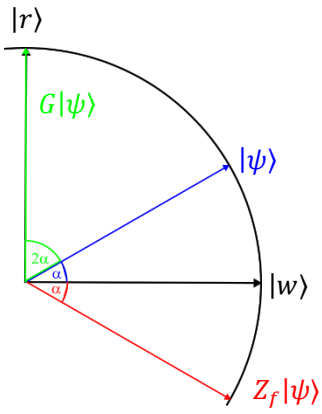
If we assume that the desired number is r = 2 ≙ 10, this can be written as
$|\psi\rangle = \frac{1}{2}|r\rangle + \frac{\sqrt{3}}{2}|w\rangle$
with
$|w\rangle = \frac{1}{\sqrt{3}}(|00\rangle + |01\rangle + |11\rangle)$.

The angle α can be calculated by the arcsine of the factor of the desired state: $\alpha = \sin^{-1}\left(\frac{1}{2}\right) = 30°$

Step II: Apply the oracle gate Z_f to $|\psi\rangle$. This gate does exactly one query on the function f. As declined in the previous section, it adds a minus to the state $|r\rangle$ and does not change any other state, which means that the vector $|\psi\rangle$ is reflected at the vector $|w\rangle$.



Step III: Apply a combination of gates, which lead to a reflection of the vector $Z_f|\psi\rangle$ at the vector $|\psi\rangle$.
The combination of step II and step III is called Grover operation G. This means, that the resulting vector is rotated by 2α in direction of the desired state in comparison to the initial state $|\psi\rangle$.



In the example N = 4 we have reached the desired state exactly with one step of the Grover operation. In general, the Grover operation has to be repeated as in the next step.

Step IV: Repeat the Grover operation (step II and step III) subsequently t times (so the number of queries on f is t) until the state vector is at most parallel to $|r\rangle$ and do a measurement on all n qubits.

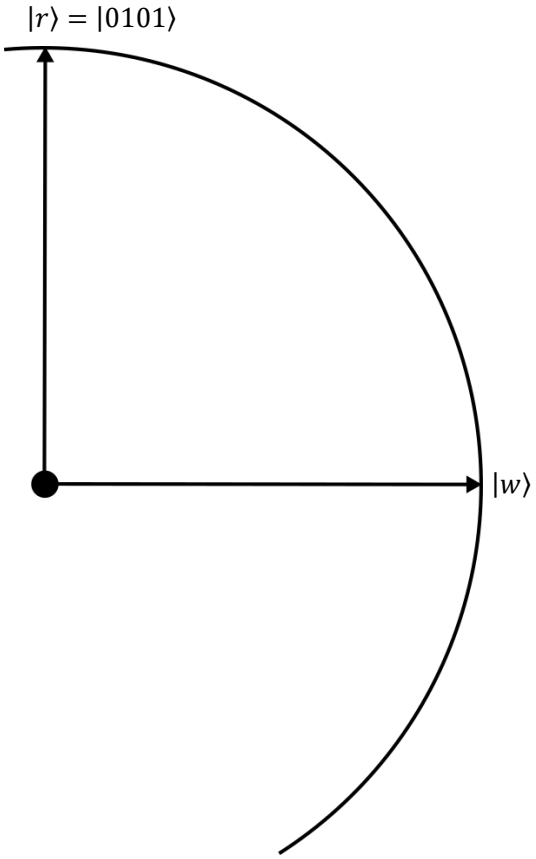
Exercise 3: Understanding Grover's Algorithm step by step

Now, you will delve into each step of the algorithm and get an understanding of how often the Grover operation must be repeated.

In this case we set $n = 4$ and $N = 2^4 = 16$.

Step I: Applying the Hadamard operation on all four qubits of the state $|0000\rangle$ gives $|\psi\rangle = \frac{1}{\sqrt{2^4}}(|0000\rangle + |0001\rangle + \ldots + |1111\rangle)$. Let us assume that $r = 5 \triangleq 0101$. So $|\psi\rangle = \frac{1}{\sqrt{2^4}}|0101\rangle + a|w\rangle$, whereby a is some normalization value.

Calculate the angle α from the factor $\frac{1}{\sqrt{2^4}} = \frac{1}{4}$ and draw the corresponding vector $|\psi\rangle$ in the picture:



Step II: Draw the state vector $Z_f|\psi\rangle$ by reflection of $|\psi\rangle$ at $|w\rangle$.

Step III: Draw the state vector $G|\psi\rangle$ by reflection of $Z_f|\psi\rangle$ at $|\psi\rangle$.

Step IV: Repeat step II and step III (so $t = 2$).

Exercise 4: Reflecting on your solution

a. Do you think that two repetitions of the Grover operation are sufficient? What would happen if you did more repetitions?

b. The Grover algorithm gives the right answer with high probability and not necessarily exact. Explain this with your solution.

c. How does the angle α behave if n increases? What does this mean for the needed number of repetitions?

Conclusions

Why does Grover only need $O(\sqrt{N})$ steps?

In general, the angle α is equal to $\sin^{-1}\left(\frac{1}{\sqrt{N}}\right)$, which is approximately equal to $\frac{1}{\sqrt{N}}$.

After one Grover operation the current angle is 3α, after the second Grover operation the angle is 5α. For t Grover operations the angle is $(2t+1)\cdot\alpha$.

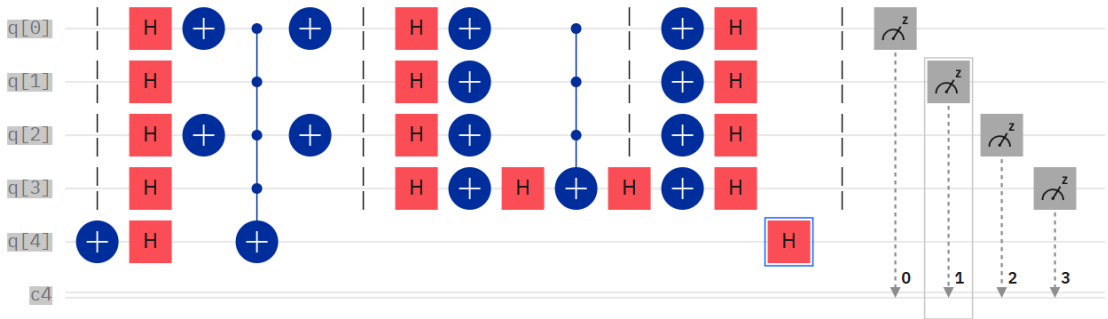
To get the right state with a high probability this angle should be close to $\frac{\pi}{2}$.
So with $(2t+1)\cdot\frac{1}{\sqrt{N}} \approx \frac{\pi}{2}$ we get $t \approx \frac{1}{2}\cdot\left(\sqrt{N}\cdot\frac{\pi}{2}-1\right)$, which is the optimal number of iterations in the Grover algorithm.

How can I be sure to get the right answer?

The Grover algorithm will in most cases give the result with a high probability and not necessarily exact. To be sure, to get the right answer, it is easy to check the result just by putting the result in the oracle. If the result is not correct, you can just run the Grover operation a second time to get another result to check.

Exercise 5: Try the Grover algorithm in the Composer

The circuit shows the 4bit Gover operation with the desired state $|r\rangle = |0101\rangle$.



Preparation | Oracle | Reflection at $|\psi\rangle$

a. Implement this circuit in the composer.
b. The circuit just implements one Grover operation. Repeat this step in the circuit and compare with your results from task 2.