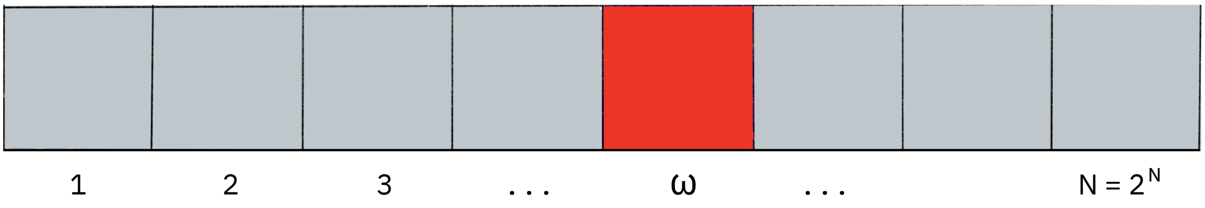
Grover's algorithm

We are now in a good place to discuss our first quantum algorithms and subroutines. Let's begin with Grover's search algorithm and the *amplitude amplification trick*.

You have likely heard that one of the many advantages a quantum computer has over a classical computer is its superior speed searching databases. Grover's algorithm demonstrates this capability. This algorithm can speed up an unstructured search problem quadratically, but its uses extend beyond that; it can serve as a general trick or subroutine to obtain quadratic run time improvements for a variety of other algorithms. This is called the *amplitude amplification trick*. But before we start the simulations, let's look at the unstructured search problem.

Unstructured search

Suppose you are given a large list of N items. Among these items is one item with a unique property that we wish to locate. We will call this one the winner, w. Think of each item in the list as a box of a particular color. Say all items in the list are gray except the winner w, which is red.



To find the red box – the *marked item* – using classical computation, one would have to check on average $N/2$ of these boxes, and in the worst case, all N of them. On a quantum computer, however, we can find the marked item in roughly $\sqrt{N}$ steps with Grover's amplitude amplification trick. It was proven (even before Grover's algorithm was discovered!) that this speedup is in fact the most we can hope for [Bennett, 1997]. A quadratic speedup is indeed a substantial time-saver for finding marked items in long lists. Additionally, the algorithm does not use the list's internal structure, which makes it *generic;* this is why it immediately provides a quadratic quantum speed-up for many classical problems.

The Oracle

How will the list items be provided to the quantum computer? For the examples in this topic, our 'database' is comprised of all the possible computational basis states our qubits can be in. For example, if we have 3 qubits, our list is the states $|000\rangle, |001\rangle, \dots |111\rangle$ (i.e the states $|0\rangle \rightarrow |7\rangle$).

Grover's algorithm solves oracles that add a negative phase to the solution states. That is, for any state $|x\rangle$ in the computational basis:

$$U_w|x\rangle = \begin{cases} |x\rangle & \text{if } x \neq w \\ -|x\rangle & \text{if } x = w \end{cases}$$

This oracle will be a diagonal matrix, where the entry that correspond to the marked item will have a negative phase. For example, if we have three qubits and $\omega = 101$, our oracle will have the matrix:

$$U_w = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{-1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

What makes Grover's algorithm so powerful is how easy it is to convert a problem to an oracle of this form. There are many computational problems in which it's difficult to find a solution, but relatively easy to verify a solution. For example, we can easily verify a solution to a sudoku by checking all the rules are satisfied. For these problems, we can create a function f that takes a proposed solution x and returns $f(x) = 0$ if x is not a solution ($x \neq w$), and $f(x) = 1$ for a valid solution ($x = w$). Our oracle can then be described as:

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

and the oracle's matrix will be a diagonal matrix of the form:

$$U_w = \begin{bmatrix} (-1)^{f(0)} & 0 & \dots & 0 \\ 0 & (-1)^{f(1)} & \dots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \dots & (-1)^{f(2^n-1)} \end{bmatrix}$$

▶ Circuit construction of a Grover oracle

For the next part of this chapter, we aim to teach the core concepts of the algorithm. We will create example oracles where we know ω beforehand, and not worry ourselves with whether these oracles are useful or not.

Amplitude amplification

So how does the algorithm work? Before looking at the list of items, we have no idea where the marked item is. Therefore, any guess of its location is as good as any other, which can be expressed in terms of a quantum state called a *uniform superposition*:

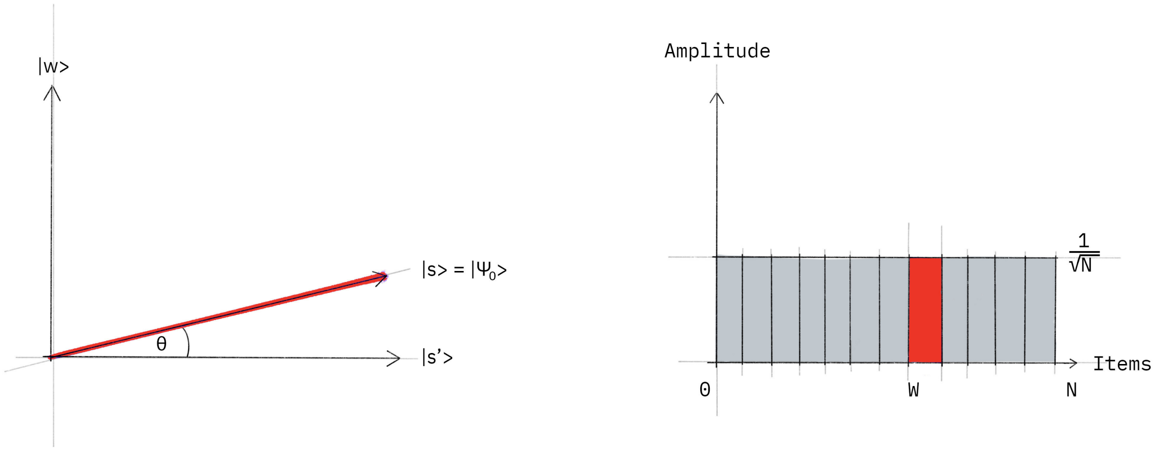
$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle.$$

If at this point we were to measure in the standard basis $\{|x\rangle\}$, this superposition would collapse, according to the fifth quantum law, to any one of the basis states with the same probability of $\frac{1}{N} = \frac{1}{2^n}$. Our chances of guessing the right value w is therefore 1 in 2^n, as could be expected. Hence, on average we would need to try about $N = 2^n$ times to guess the correct item.

Enter the *amplitude amplification* procedure, which is how a quantum computer significantly enhances this probability. This procedure stretches out (amplifies) the amplitude of the marked item, which shrinks the other items’ amplitudes, so that measuring the final state will return the right item with near certainty.

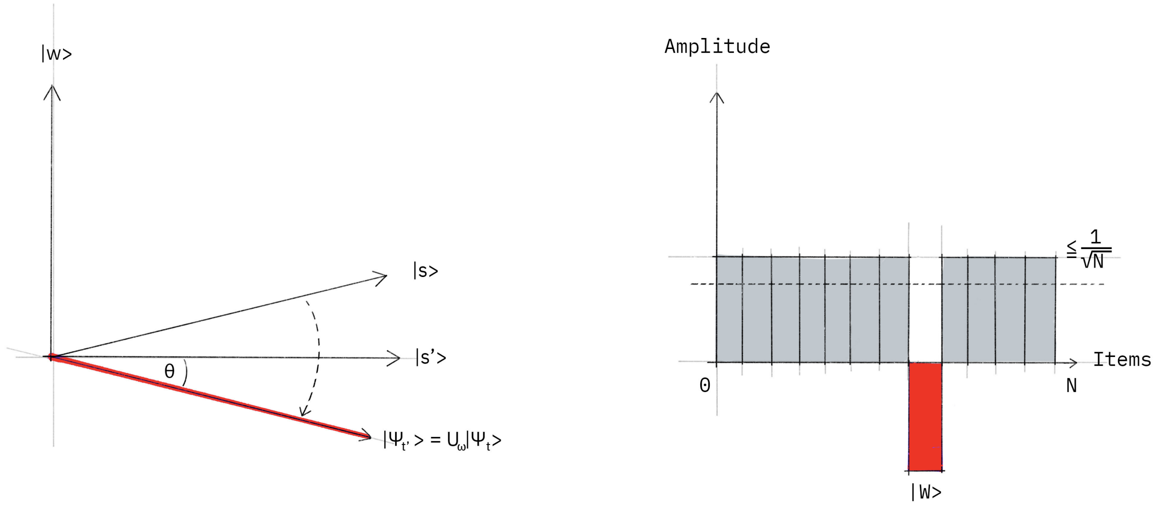
This algorithm has a nice geometrical interpretation in terms of two reflections, which generate a rotation in a two-dimensional plane. The only two special states we need to consider are the winner $|w\rangle$ and the uniform superposition $|s\rangle$. These two vectors span a two-dimensional plane in the vector space $\mathbb{C}^N$. They are not quite perpendicular because $|w\rangle$ occurs in the superposition with amplitude $N^{-1/2}$ as well. We can, however, introduce an additional state $|s'\rangle$ that is in the span of these two vectors, is perpendicular to $|w\rangle$, and is obtained from $|s\rangle$ by removing $|w\rangle$ and rescaling.

Step 1 The amplitude amplification procedure starts out in the uniform superposition $|s\rangle$. (The uniform superposition is easily constructed from $|s\rangle = H^{\otimes n}|0\rangle^n$, as was shown in Creating superpositions and quantum interference.)



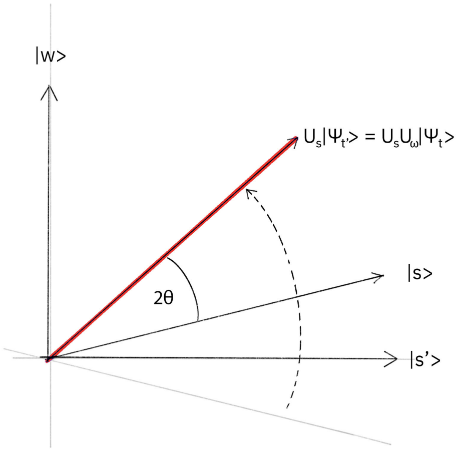
The left graphic corresponds to the two-dimensional plane spanned by perpendicular vectors $|w\rangle$ and $|s'\rangle$, which allows us to express the initial state as $|s\rangle = sin\theta|w\rangle + cos\theta|s'\rangle$, where $\theta = arcsin\langle s|w\rangle = arcsin\frac{1}{\sqrt{N}}$. The right graphic is a bar graph of the amplitudes of the state $|s\rangle$.

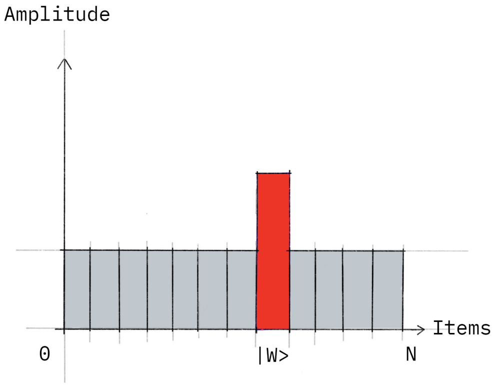
Step 2 We apply the oracle reflection U_f to the state $|s\rangle$.



Geometrically this corresponds to a reflection of the state $|s\rangle$ about $|s'\rangle$. This transformation means that the amplitude in front of the $|w\rangle$ state becomes negative, which in turn means that the average amplitude (indicated by a dashed line) has been lowered.

Step 3 We now apply an additional reflection U_s about the state $|s\rangle$: $U_s = 2|s\rangle\langle s| - 1$. This transformation maps the state to $U_s U_f |s\rangle$ and completes the transformation.

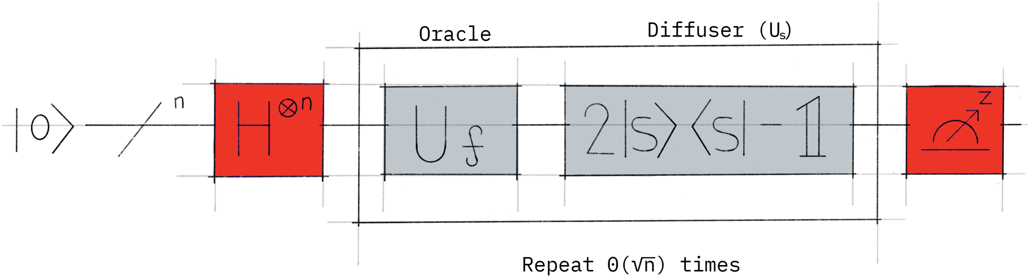




Two reflections always correspond to a rotation. The transformation $U_s U_f$ rotates the initial state $|s\rangle$ closer toward the winner $|w\rangle$. The action of the reflection U_s in the amplitude bar diagram can be understood as a reflection about the average amplitude. Since the average amplitude has been lowered by the first reflection, this transformation boosts the negative amplitude of $|w\rangle$ to roughly three times its original value, while it decreases the other amplitudes. We then go to **Step 2** to repeat the application. This procedure will be repeated several times to focus in on the winner.

After t steps, the state will have transformed to $|\psi_t\rangle$, where $|\psi_t\rangle = (U_s U_f)^t |s\rangle$. How many times do we need to apply the rotation? It turns out that roughly $\sqrt{N}$ rotations suffice. This becomes clear when looking at the amplitudes of the state $|\psi\rangle$. We can see that the amplitude of $|w\rangle$ grows linearly with the number of applications $\sim tN^{-1/2}$. However, since we are dealing with amplitudes and not probabilities, the vector space’s dimension enters as a square root. Therefore it is the amplitude, and not just the probability, that is being amplified in this procedure.

In the case that there are multiple solutions, M, it can be shown that roughly $\sqrt{\frac{N}{M}}$ rotations will suffice.



Example: two qubits

Let us now examine a simple example. The smallest circuit for which this can be implemented involves two qubits, that is, $N = 2^2$. First, we will determine how many rotations are required to rotate the initial state $|s\rangle$ to the winner $|w\rangle[3]$.

1. Following the previously described steps, for the case $N = 4$, we have

$$\theta = arcsin\frac{1}{2} = \frac{\pi}{6}$$

2. After t steps, we have

$$(U_sU_w)^t|s\rangle = sin\theta_t|w\rangle + cos\theta_t|s'\rangle$$

where

$$\theta_t = (2t+1)\theta$$

3. In order to obtain $|w\rangle$, we need $\theta_t = \frac{\pi}{2}$. Substituting $\theta = \frac{\pi}{6}$ into the above gives us $t = 1$. Therefore, after $t = 1$ rotation, the desired element is found.

Oracle for $|w\rangle = |11\rangle$

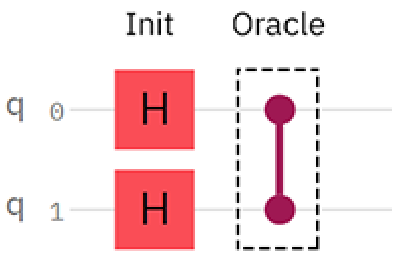
We will review the case $|w\rangle = |11\rangle$. The oracle U_w in this case acts as follows:

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

or:

$$U_w = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

which you may recognise as the controlled-Z gate. Thus, for this example, our oracle is simply the controlled-Z gate:



Reflection U_s

To complete the circuit, we need to implement the additional reflection $U_s = 2|s\rangle\langle s| - 1$. Since this is a reflection about $|s\rangle$, we want to add a negative phase to every state orthogonal to $|s\rangle$.

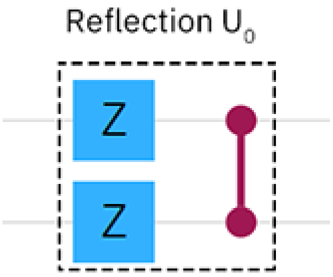
One way to do this is to use the operation that transforms the state $|s\rangle \rightarrow |0\rangle$, which we know is the Hadamard gate applied to each qubit:

$$H^{\otimes n}|s\rangle = |0\rangle$$

Then we apply a circuit that adds a negative phase to the states orthogonal to $|0\rangle$:

$$U_0 \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \frac{1}{2}(|00\rangle - |01\rangle - |10\rangle - |11\rangle)$$

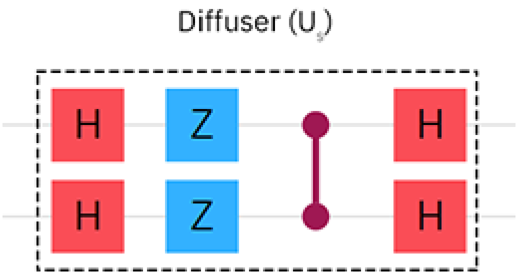
One way of implementing U_0 is the following circuit:



Finally, we perform the operation that transforms the state $|0\rangle \rightarrow |s\rangle$ (the Hadamard gate) again:

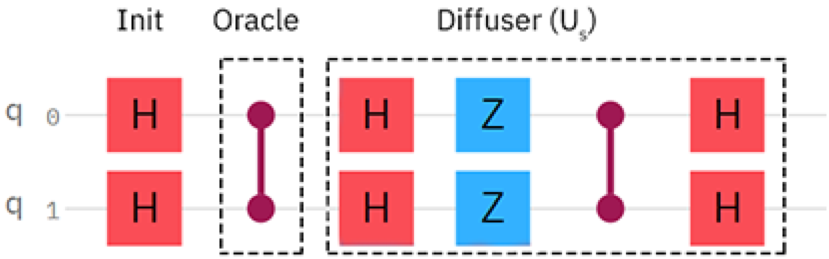
$$H^{\otimes n} U_0 H^{\otimes n} = U_s$$

The complete circuit for U_s looks like this:



Full Circuit for $|w\rangle = |11\rangle$

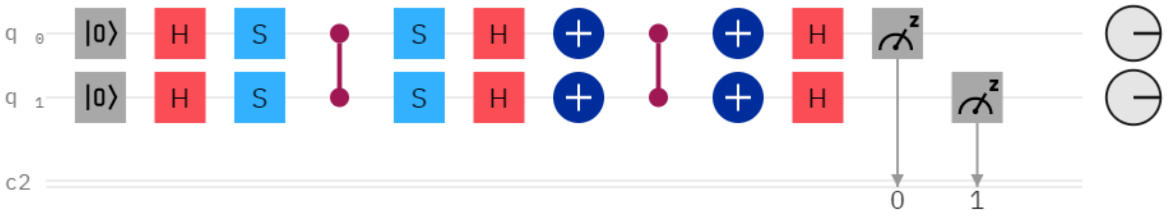
Since in the case of $N = 4$, only one rotation is required, we can combine the above components to build the full circuit for Grover's algorithm for the case $|w\rangle = |11\rangle$:



Use IBM Quantum Composer to explore the oracles

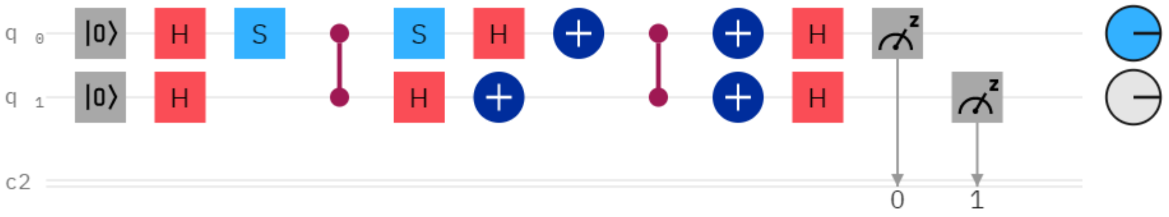
There are four possible oracles U_f, one for each choice of the winner. In each of the following examples, we first create the uniform superposition, then tag the state with U_f, and finally perform U_s.

Grover $|w\rangle = |00\rangle$



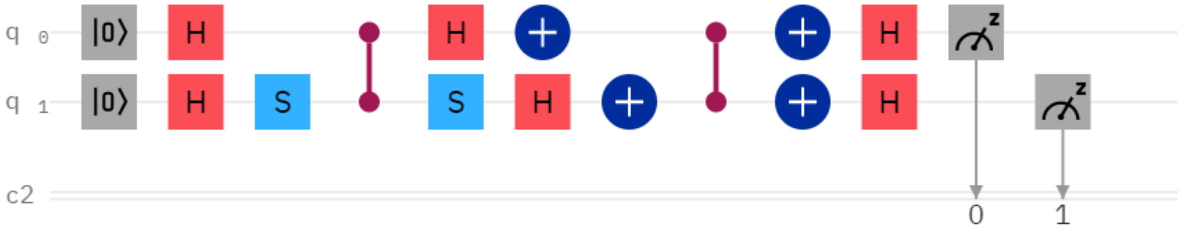
Open in IBM Quantum Composer

Grover $|w\rangle = |01\rangle$



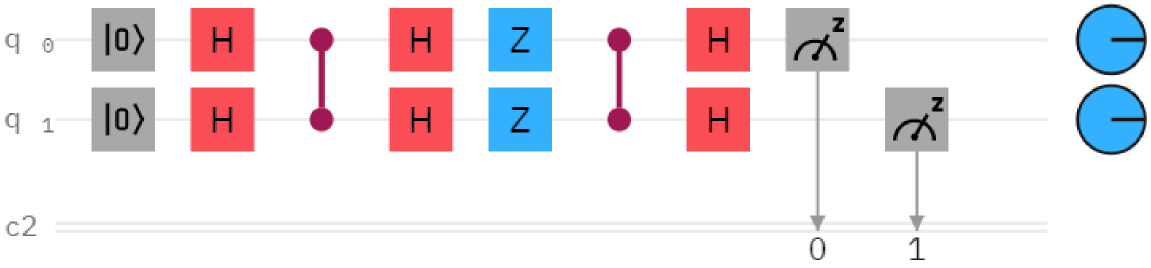
Open in IBM Quantum Composer

Grover $|w\rangle = |10\rangle$



Open in IBM Quantum Composer

Grover $|w\rangle = |11\rangle$



Open in IBM Quantum Composer

Next: Deutsch-Jozsa algorithm

Previous: Entanglement