Quantum Computing
5 – Simple quantum algorithms

Angelo Bassi

Algorithms with oracle

Suppose we are supplied with a quantum **oracle** – a black box whose internal workings are not important at this stage – with the ability to recognize solutions to a given problem by computing a suitable function f. More precisely, the oracle is a unitary operator, U_f, defined by its action on the computational basis:

$$U_f \,:\, |x, y\rangle \,\mapsto\, |x, y \oplus f(x)\rangle$$

where $\oplus$ is addition mod 2.

Suppose U_f acts on the input which is a **superposition of many states**. Since U_f is a linear operator, it acts simultaneously on all the vectors that constitute the superposition. **Thus the output is also a superposition of all the results**

$$U_f : \sum_x |x\rangle|0\rangle \mapsto \sum_x |x\rangle|f(x)\rangle.$$

All values of the function computed at once. Very easy!!
But... **measurements will make the wave function collapse giving only one output**. No advantage.

The goal of a quantum algorithm is to operate in such a way that the particular outcome we want to observe has a larger probability to be measured than the other outcomes.

Deutsch Algorithm

Let $f : \{0,1\} \to \{0,1\}$ be a binary function. Note that there are only four possible f, namely

$$f_1 : 0 \mapsto 0,\ 1 \mapsto 0, \quad f_2 : 0 \mapsto 1,\ 1 \mapsto 1,$$
$$f_3 : 0 \mapsto 0,\ 1 \mapsto 1, \quad f_4 : 0 \mapsto 1,\ 1 \mapsto 0.$$

The first two cases, f_1 and f_2, are called *constant*, while the rest, f_3 and f_4, are *balanced*. If we only have classical resources, we need to evaluate f twice to tell if f is constant or balanced. There is a quantum algorithm, however, with which it is possible to tell if f is constant or balanced with a single evaluation of f, as was shown by Deutsch [2].

First we need to turn the classical function f(x) into a quantum one.
To this purpose we define the quantum oracle

$$U_f \ : \ |x, y\rangle \mapsto |x, y \oplus f(x)\rangle$$

The algorithm is structured as follows.

1. Start with the state $|01\rangle$.
2. Apply an Hadamard on both qubits: $\frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle)$
3. Apply the operator U_f implementing the function

$$\frac{1}{2}(|0, f(0)\rangle - |0, 1 \oplus f(0)\rangle + |1, f(1)\rangle - |1, 1 \oplus f(1)\rangle)$$
$$= \frac{1}{2}(|0, f(0)\rangle - |0, \neg f(0)\rangle + |1, f(1)\rangle - |1, \neg f(1)\rangle),$$

Quantum parallelism: all values computed at once

Deutsch Algorithm

4. Apply an Hadamard to the first qubit

$$\frac{1}{2\sqrt{2}}\left[(|0\rangle+|1\rangle)(|f(0)\rangle-|\neg f(0)\rangle)+(|0\rangle-|1\rangle)(|f(1)\rangle-|\neg f(1)\rangle)\right]$$

If the function is constant, the two blue terms are equal, the state reduces to

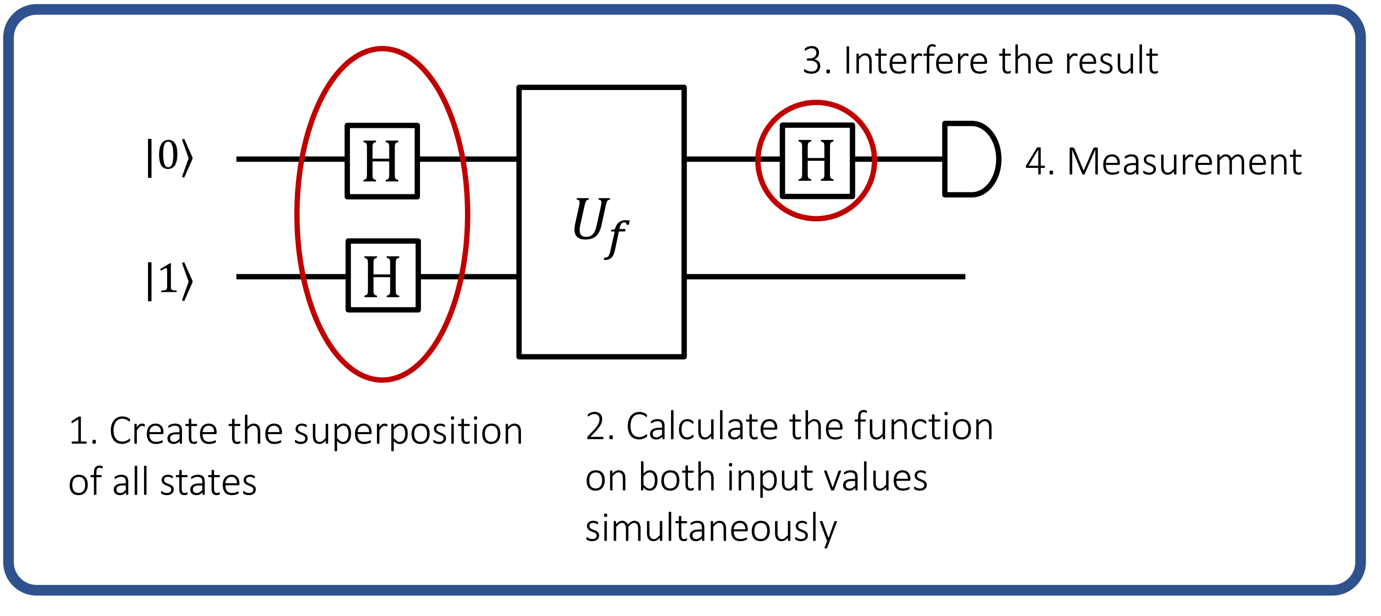
$$\frac{1}{\sqrt{2}}|0\rangle(|f(0)\rangle-|\neg f(0)\rangle)$$

If the function is balanced, the two blue terms are opposite, the state reduces to

$$\frac{1}{\sqrt{2}}|1\rangle(|f(0)\rangle-|\neg f(0)\rangle)$$

Therefore **the measurement of the first qubit tells us whether f is constant or balanced.**

5. Measure the first qubit to determine f. The full algorithm is:



Deutsch-Jozsa Algorithm

Let us first define the **Deutsch-Jozsa problem**. Suppose there is a binary function

$$f : S_n \equiv \{0, 1, \ldots, 2^n - 1\} \to \{0, 1\}.$$

We require that f be either *constant* or *balanced* as before. When f is constant, it takes a constant value 0 or 1 irrespetive of the input value x. When it is balanaced the value $f(x)$ for the half of $x \in S_n$ is 0, while it is 1 for the rest of x. In other words, $|f^{-1}(0)| = |f^{-1}(1)| = 2^{n-1}$, where $|A|$ denotes the number

It is clear that we need at least $2^{n-1} + 1$ steps, in the worst case with classical manipulations, to make sure if $f(x)$ is constant or balanced with 100% confidence. It will be shown below that the number of steps reduces to a single step if we are allowed to use a quantum algorithm.

The **oracle** for the Deutsch-Jozsa algorithm is always the same

$$U_f : |x, y\rangle \mapsto |x, y \oplus f(x)\rangle$$

1. Prepare an $(n+1)$-qubit register in the state $|\psi_0\rangle = |0\rangle^{\otimes n} \otimes |1\rangle$. First n qubits work as input qubits, while the $(n+1)$st qubit serves as a "scratch pad." Such qubits, which are neither input qubits nor output qubits, but work as a scratch pad to store temporary information are called **ancillas** or **ancillary qubits**.

2. Apply the Walsh-Hadamard transforamtion to the register. Then we have the state

$$U_{\mathrm{H}}^{\otimes n+1}|\psi_0\rangle = \frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle)^{\otimes n} \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$
$$= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

3. Apply the oracle. The state changes into

$$= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \frac{1}{\sqrt{2}}(|f(x)\rangle - |\neg f(x)\rangle)$$
$$= \frac{1}{\sqrt{2^n}} \sum_{x} (-1)^{f(x)} |x\rangle \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

Although the gate U_f is applied once for all, it is applied to *all* the n-qubit states $|x\rangle$ simultaneously.

Deutsch-Jozsa Algorithm

4. The Walsh-Hadamard transformation is applied on the first n qubits next. We obtain

$$\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}(-1)^{f(x)}U_{\mathrm{H}}^{\otimes n}|x\rangle\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle$$

On the Hadamard gate

It is instructive to write the action of the one-qubit Hadamard gate in the following form,

$$U_{\mathrm{H}}|x\rangle=\frac{1}{\sqrt{2}}(|0\rangle+(-1)^x|1\rangle)=\frac{1}{\sqrt{2}}\sum_{y\in\{0,1\}}(-1)^{xy}|y\rangle,$$

where $x\in\{0,1\}$, to find the resulting state. The action of the Walsh-Hadamard transformation on $|x\rangle=|x_{n-1}\ldots x_1x_0\rangle$ yields

$$\begin{aligned}W_n|x\rangle &= (U_{\mathrm{H}}|x_{n-1}\rangle)(U_{\mathrm{H}}|x_{n-2}\rangle)\ldots(U_{\mathrm{H}}|x_0\rangle)\\ &= \frac{1}{\sqrt{2^n}}\sum_{y_{n-1},y_{n-2},\ldots,y_0\in\{0,1\}}(-1)^{x_{n-1}y_{n-1}+x_{n-2}y_{n-2}+\ldots+x_0y_0}\\ &\quad\times|y_{n-1}y_{n-2}\cdots y_0\rangle\\ &= \frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1}(-1)^{x\cdot y}|y\rangle,\end{aligned}$$

where $x\cdot y=x_{n-1}y_{n-1}\oplus x_{n-2}y_{n-2}\oplus\ldots\oplus x_0y_0$.

We then have

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

Deutsch-Jozsa Algorithm

5. The first n qubits are measured. Suppose $f(x)$ is constant. Then

$$\frac{1}{2^n}\sum_{x,y}(-1)^{x\cdot y}|y\rangle\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)$$

up to an overall phase. Now let us consider the summation

$$\frac{1}{2^n}\sum_{x=0}^{2^n-1}(-1)^{x\cdot y}$$

with a fixed $y \in S_n$. Clearly it vanishes since $x \cdot y$ is 0 for half of x and 1 for the other half of x unless $y = 0$. Therefore the summation yields δ_{y0}. Now the state reduces to

$$|0\rangle^{\otimes n}\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle),$$

and the measurement outcome of the first n qubits is always $00\ldots0$.

Suppose $f(x)$ is balanced next. The probability amplitude of $|y=0\rangle$ in $|\psi_3\rangle$ is proportional to

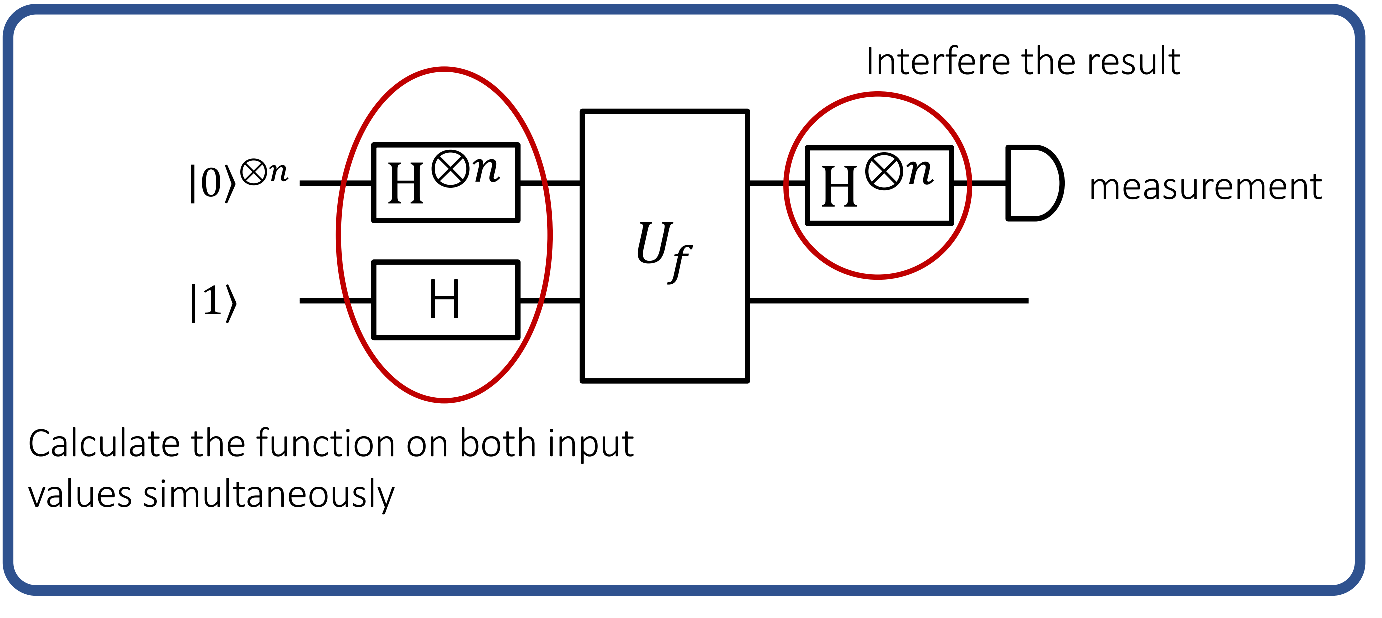
$$\sum_{x=0}^{2^n-1}(-1)^{f(x)}(-1)^{x\cdot 0}=\sum_{x=0}^{2^n-1}(-1)^{f(x)}=0.$$

Therefore the probability of obtaining measurement outcome $00\ldots0$ for the first n qubits vanishes. In conclusion, the function f is constant if we obtain $00\ldots0$ upon the meaurement of the first n qubits in the state $|\psi_3\rangle$, and it is balanced otherwise.

Example with 3 qubits.

Take y = 110. Then $x \cdot y = x_2 \oplus x_1$

x	$x_2 \oplus x_1$
000	0
001	0
010	1
011	1
100	1
101	1
110	0
111	0



Bernstein-Vazirani Algorithm

The **Bernstein-Vazirani algorithm** is a special case of the Deutsch-Jozsa algorithm, in which $f(x)$ is given by $f(x) = c \cdot x$, where $c = c_{n-1}c_{n-2}\dots c_0$ is an n-bit binary number [4]. Our aim is to find c with the smallest number of evaluations of f. If we apply the Deutsch-Jozsa algorithm with this f, we obtain

$$|\psi_3\rangle = \frac{1}{2^n}\left[\sum_{x,y=0}^{2^n-1}(-1)^{c\cdot x}(-1)^{x\cdot y}|y\rangle\right]\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

Let us fix y first. If we take $y = c$, we obtain

$$\sum_x (-1)^{c\cdot x}(-1)^{x\cdot c} = \sum_x (-1)^{2c\cdot x} = 2^n.$$

If $y \neq c$, on the other hand, there will be the same number of x such that $c \cdot x = 0$ and x such that $c \cdot x = 1$ in the summation over x and, as a result, the probability amplitude of $|y \neq c\rangle$ vanishes. By using these results, we end up with

$$|\psi_3\rangle = |c\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

We are able to tell what c is by measuring the first n qubits.

Exercise

EXERCISE 5.1 Let us take $n = 2$ for definiteness. Consider the following cases and find the final wave function $|\psi_3\rangle$ and evaluate the measurement outcomes and their probabilities for each case.
(1) $f(x) = 1\ \forall x \in S_2$.
(2) $f(00) = f(01) = 1, f(10) = f(11) = 0$.
(3) $f(00) = 0, f(01) = f(10) = f(11) = 1$. (This function is neither constant nor balanced.)

EXERCISE 5.2 Consider the Bernstein-Vazirani algorithm with $n = 3$ and $c = 101$. Work out the quantum circuit depicted in Fig. 5.2 to show that the measurement outcome of the first three qubits is $c = 101$.

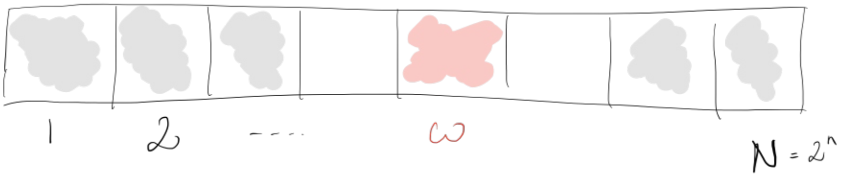
Grover's search algorithm

Suppose there is a stack of $N = 2^n$ files, randomly placed, that are numbered by $x \in S_n \equiv \{0, 1, \ldots, N-1\}$. Our task is to find an algorithm which picks out a particular file which satisfies a certain condition.

In mathematical language, this is expressed as follows. Let $f : S_n \to \{0, 1\}$ be a function defined by

$$f(x) = \begin{cases} 1 & (x = z) \\ 0 & (x \neq z), \end{cases}$$

where z is the address of the file we are looking for. It is assumed that $f(x)$ is *instantaneously* calculable, such that this process does not require any computational steps. A function of this sort is often called an oracle as noted in Chapter 5. Thus, the problem is to find z such that $f(z) = 1$, given a function $f : S_n \to \{0, 1\}$ which assumes the value 1 only at a single point.



Clearly we have to check one file after another in a classical algorithm, and it will take $O(N)$ steps on average. It is shown below that it takes only $O(\sqrt{N})$ steps with Grover's algorithm. This is accomplished by *amplifying* the amplitude of the vector $|z\rangle$ while cancelling that of the vectors $|x\rangle$ $(x \neq z)$.

We first needs to implement the function f(x) quantum mechanically. The **oracle** U_f is defined in the usual way

$$U_f|x, y\rangle = |x, y \oplus f(x)\rangle \qquad \text{with } f(x) = \text{either 0 or 1}$$

In particular we have

$$U_f|x\rangle \frac{1}{\sqrt{2}}[|0\rangle - |1\rangle] = (-1)^{f(x)}|x\rangle \frac{1}{\sqrt{2}}[|0\rangle - |1\rangle]$$

Grover’s search algorithm

Therefore, without loss of generality, we can neglect the last qubit and assume

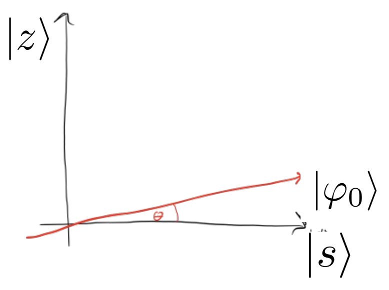
$$U_f|x\rangle = (-1)^{f(x)}|x\rangle$$

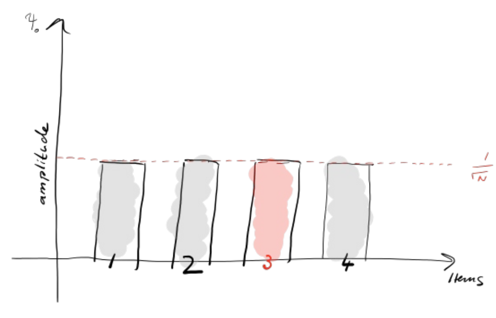
on the computational basis. We see that if x is an unmarked item, the oracle does nothing to the state. It flips the phase for the marked item. It is easy to see that

$$U_f = I - 2|z\rangle\langle z|$$

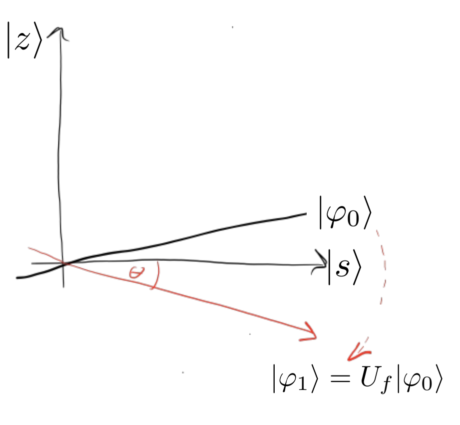
Step 1: Create an initially **equal weighted superposition** of all states (this is done with N Hadamard gates):

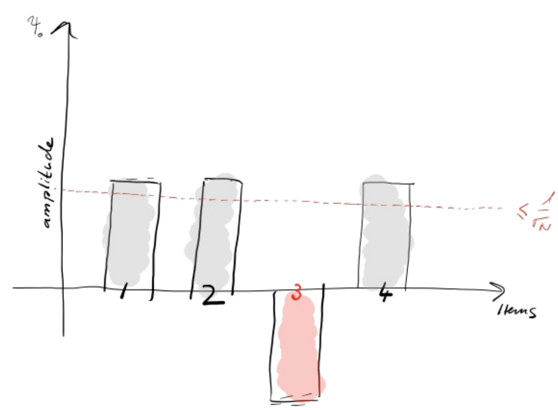
$$|\varphi_0\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle.$$



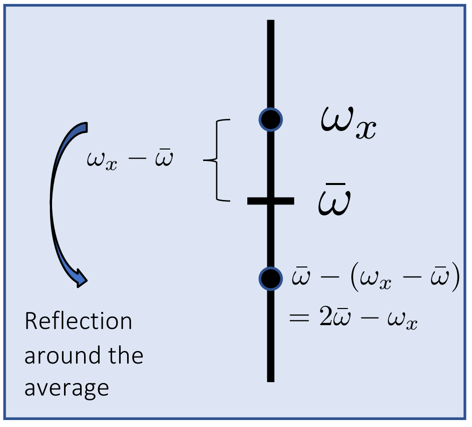


Step 2: Apply the oracle U_f. Geometrically this corresponds to a reflection of the state $|z\rangle$ about $|s\rangle$. This transformation means that the amplitude in front of the $|z\rangle$ state becomes negative, which in turn means that the average amplitude has been lowered.





Grover's search algorithm



Step 3: Apply the gate

$$D = -I + 2|\varphi_0\rangle\langle\varphi_0|.$$

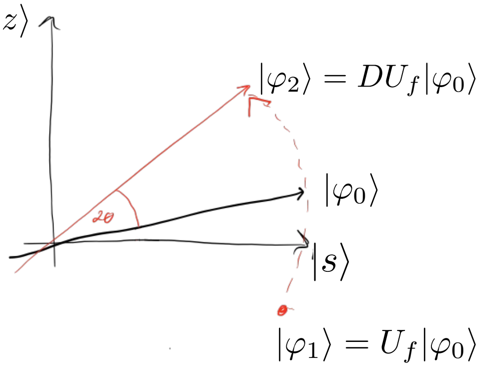
The action of the gate is the following

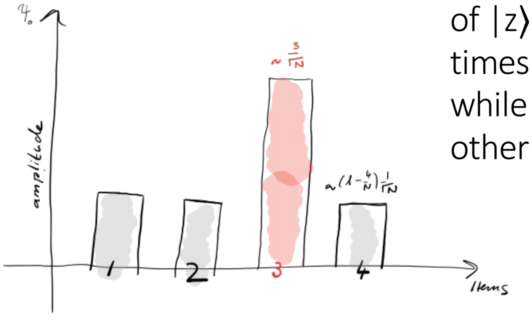
$$|\varphi\rangle = \sum_{x=0}^{N-1} \omega_x |x\rangle \;\rightarrow\; D|\varphi\rangle = \left[\frac{2}{N}\sum_{x,y=0}^{N-1} |x\rangle\langle y|\right] \sum_{z=0}^{N-1} \omega_z |z\rangle - \sum_{x=0}^{N-1} \omega_x |x\rangle$$

$$= \frac{2}{N}\left[\sum_{x=0}^{N-1} |x\rangle\right]\left[\sum_{y=0}^{N-1} \omega_y\right] - \sum_{x=0}^{N-1} \omega_x |x\rangle = \sum_{x=0}^{N-1} (2\bar{\omega} - \omega_x)|x\rangle$$

with $\bar{\omega} = \frac{1}{N}\sum_{x=0}^{N-1} \omega_x$ average

In our case we get





Since the average amplitude has been lowered by the first reflection, this transformation boosts the negative amplitude of |z⟩ to roughly three times its original value, while it decreases the other amplitudes.

Grover's search algorithm

Step 3: go to step 2 an repeat the application of U_f and D a sufficient number of times. Let us call $G_f = D\, U_f$.

PROPOSITION 7.2 Let us write

$$G_f^k|\varphi_0\rangle = a_k|z\rangle + b_k \sum_{x\neq z} |x\rangle$$

with the initial condition

$$a_0 = b_0 = \frac{1}{\sqrt{N}}.$$

Then the coefficients $\{a_k, b_k\}$ for $k \geq 1$ satisfy the recursion relations

$$a_k = \frac{N-2}{N} a_{k-1} + \frac{2(N-1)}{N} b_{k-1}, \tag{7.18}$$

$$b_k = -\frac{2}{N} a_{k-1} + \frac{N-2}{N} b_{k-1} \tag{7.19}$$

for $k = 1, 2, \ldots$.

Proof. It is easy to see the recursion relations are satified for $k = 1$

Let $G_f^{k-1}|\varphi_0\rangle = a_{k-1}|z\rangle + b_{k-1}\sum_{x\neq z}|x\rangle$. Then

$$\begin{aligned}
G_f^k|\varphi_0\rangle =& G_f\left(a_{k-1}|z\rangle + b_{k-1}\sum_{x\neq z}|x\rangle\right) \\
&= (-I + 2|\varphi_0\rangle\langle\varphi_0|)\left(-a_{k-1}|z\rangle + b_{k-1}\sum_{x\neq z}|x\rangle\right) \\
&= -b_{k-1}\sum_{x\neq z}|x\rangle + a_{k-1}|z\rangle + \frac{2}{\sqrt{N}}(N-1)b_{k-1}|\varphi_0\rangle - \frac{2a_{k-1}}{\sqrt{N}}|\varphi_0\rangle \\
&= -b_{k-1}\sum_{x\neq z}|x\rangle + a_{k-1}|z\rangle + \frac{2}{N}(N-1)b_{k-1}\sum_x|x\rangle - \frac{2a_{k-1}}{N}\sum_x|x\rangle \\
&= \left[\frac{N-2}{N}a_{k-1} + \frac{2(N-1)}{N}b_{k-1}\right]|z\rangle + \left[-\frac{2}{N}a_{k-1} + \frac{N-2}{N}b_{k-1}\right]\sum_{x\neq z}|x\rangle,
\end{aligned}$$

and proposition is proved. ∎

Grover's search algorithm

PROPOSITION 7.3 The solutions of the recursion relations in Proposition 7.2 are explicitly given by

$$a_k = \sin[(2k+1)\theta], \quad b_k = \frac{1}{\sqrt{N-1}}\cos[(2k+1)\theta], \tag{7.20}$$

for $k = 0, 1, 2, \ldots$, where

$$\sin\theta = \sqrt{\frac{1}{N}}, \quad \cos\theta = \sqrt{1-\frac{1}{N}}. \tag{7.21}$$

Proof. Let $c_k = \sqrt{N-1}b_k$. The recursion relations (7.18) and (7.19) are written in a matrix form,

$$\begin{pmatrix} a_k \\ c_k \end{pmatrix} = M\begin{pmatrix} a_{k-1} \\ c_{k-1} \end{pmatrix}, \; M = \begin{pmatrix} (N-2)/N & 2\sqrt{N-1}/N \\ -2\sqrt{N-1}/N & (N-2)/N \end{pmatrix} = \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ -\sin 2\theta & \cos 2\theta \end{pmatrix}.$$

Note that M is a rotation matrix in $\mathbb{R}^2$, and its kth power is another rotation matrix corresponding to a rotation angle $2k\theta$. Thus the above recursion relation is easily solved to yield

$$\begin{pmatrix} a_k \\ c_k \end{pmatrix} = M^k\begin{pmatrix} a_0 \\ c_0 \end{pmatrix} = \begin{pmatrix} \cos 2k\theta & \sin 2k\theta \\ -\sin 2k\theta & \cos 2k\theta \end{pmatrix}\begin{pmatrix} \sin\theta \\ \cos\theta \end{pmatrix} = \begin{pmatrix} \sin[(2k+1)\theta] \\ \cos[(2k+1)\theta] \end{pmatrix}.$$

Replacing c_k by b_k proves the proposition. ∎

We have proved that the application of G_f k times on $|\varphi_0\rangle$ results in the state

$$G_f^k|\varphi_0\rangle = \sin[(2k+1)\theta]|z\rangle + \frac{1}{\sqrt{N-1}}\cos[(2k+1)\theta]\sum_{x\neq z}|x\rangle. \tag{7.22}$$

Measurement of the state $U_f^k|\varphi_0\rangle$ yields $|z\rangle$ with the probability

$$P_{z,k} = \sin^2[(2k+1)\theta]. \tag{7.23}$$

STEP 4 Our final task is to find the k that maximizes $P_{z,k}$. A rough estimate for the maximizing k is obtained by putting

$$(2k+1)\theta = \frac{\pi}{2} \to k = \frac{1}{2}\left(\frac{\pi}{2\theta}-1\right). \tag{7.24}$$

Grover's search algorithm

PROPOSITION 7.4 Let $N \gg 1$ and let

$$m = \left\lfloor \frac{\pi}{4\theta} \right\rfloor, \tag{7.25}$$

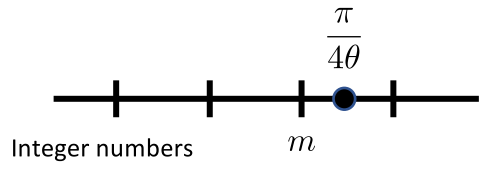
where $\lfloor x \rfloor$ stands for the floor of x. The file we are searching for will be obtained in $U_f^m|\varphi_0\rangle$ G_f^m with the probability

$$P_{z,m} \geq 1 - \frac{1}{N} \tag{7.26}$$

and

$$m = O(\sqrt{N}). \tag{7.27}$$

This is the number of times we repeat the algorithm, which grows with the square root of N



Proof. Equation (7.25) leads to the inequality $\pi/4\theta - 1 < m \leq \pi/4\theta$. Let us define $\tilde{m}$ by

$$(2\tilde{m}+1)\theta = \frac{\pi}{2} \to \tilde{m} = \frac{\pi}{4\theta} - \frac{1}{2}.$$

Observe that m and $\tilde{m}$ satisfy

$$|m - \tilde{m}| \leq \frac{1}{2}, \tag{7.28}$$

from which it follows that

$$|(2m+1)\theta - (2\tilde{m}+1)\theta| = \left|(2m+1)\theta - \frac{\pi}{2}\right| \leq \theta. \tag{7.29}$$

Considering that $\theta \sim 1/\sqrt{N}$ is a small number when $N \gg 1$ and $\sin x$ is monotonically increasing in the neighborhood of $x = 0$, we obtain

$$0 < \sin|(2m+1)\theta - \pi/2| < \sin\theta$$

$= \cos[(2m+1)\theta]$

or

$$\cos^2[(2m+1)\theta] \leq \sin^2\theta = \frac{1}{N}. \tag{7.30}$$

Thus it has been shown that

$$P_{m,z} = \sin^2[(2m+1)\theta] = 1 - \cos^2[(2m+1)\theta] \geq 1 - \frac{1}{N}. \tag{7.31}$$

It also follows from $\theta > \sin\theta = 1/\sqrt{N}$ that

$$m = \left\lfloor \frac{\pi}{4\theta} \right\rfloor \leq \frac{\pi}{4\theta} \leq \frac{\pi}{4}\sqrt{N}. \tag{7.32}$$

∎

Grover's search algorithm

It is important to note that this quantum algorithm takes only $O(\sqrt{N})$ steps and this is much faster than the classical counterpart which requires $O(N)$ steps.

We now show how to implement the gates

Selective Phase Rotation Transform

DEFINITION 6.2 (**Selective Phase Rotation Transform**) Let us define a kernel

$$K_n(x,y) = e^{i\theta_x}\delta_{xy}, \quad \forall x, y \in S_n, \tag{6.43}$$

Selective phase rotation is then defined by the following unitary operator

$$U|x\rangle = \sum_{y=0}^{N-1} K(y,x)|y\rangle.$$

The matrix representations for K_1 and K_2 are

$$K_1 = \begin{pmatrix} e^{i\theta_0} & 0 \\ 0 & e^{i\theta_1} \end{pmatrix}, \quad K_2 = \begin{pmatrix} e^{i\theta_0} & 0 & 0 & 0 \\ 0 & e^{i\theta_1} & 0 & 0 \\ 0 & 0 & e^{i\theta_2} & 0 \\ 0 & 0 & 0 & e^{i\theta_3} \end{pmatrix}.$$

Selective Phase Rotation Transform

The implementation of K_n is achieved with the universal set of gates as follows. Take $n = 2$, for example. The kernel K_2 has been given above. This is decomposed as a product of two two-level unitary matrices as

$$K_2 = A_0 A_1, \tag{6.45}$$

where

$$A_0 = \begin{pmatrix} e^{i\theta_0} & 0 & 0 & 0 \\ 0 & e^{i\theta_1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, A_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\theta_2} & 0 \\ 0 & 0 & 0 & e^{i\theta_3} \end{pmatrix}. \tag{6.46}$$

Note that

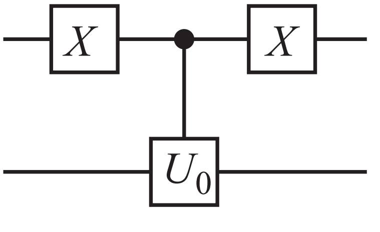
$$A_0 = |0\rangle\langle 0| \otimes U_0 + |1\rangle\langle 1| \otimes I, \quad U_0 = \begin{pmatrix} e^{i\theta_0} & 0 \\ 0 & e^{i\theta_1} \end{pmatrix},$$
$$A_1 = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U_1, \quad U_1 = \begin{pmatrix} e^{i\theta_2} & 0 \\ 0 & e^{i\theta_3} \end{pmatrix}.$$

Thus A_1 is realized as an ordinary controlled-U_1 gate while the control bit is negated in A_0. Then what we have to do for A_0 is to negate the control bit first and then to apply ordinary controlled-U_0 gate and finally to negate the control bit back to its input state. In summary, A_0 is implemented as in

Fig. 6.5. In fact, it can be readily verified that the gate in Fig. 6.5 is written as

$$\begin{aligned}&(X \otimes I)(|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U_0)(X \otimes I) \\ &= X|0\rangle\langle 0|X \otimes I + X|1\rangle\langle 1|X \otimes U_0 = |1\rangle\langle 1| \otimes I + |0\rangle\langle 0| \otimes U_0 = A_0.\end{aligned}$$

Thus these gates are implemented with the set of universal gates. In fact, the order of A_i does not matter since $[A_0, A_1] = 0$.



Back on Grover's search algorithm

We need to prove that the D gate used to perform the quantum search can be implemented efficiently. We now show that

$$D = W_n R_0 W_n, \tag{7.6}$$

where W_n is the Walsh-Hadamard transform,

$$W_n(x,y) = \frac{1}{\sqrt{N}}(-1)^{x\cdot y}, \quad (x, y \in S_n) \tag{7.7}$$

and R_0 is the selective phase rotation transform defined by

$$R_0(x,y) = e^{i\pi(1-\delta_{x0})}\delta_{xy} = (-1)^{1-\delta_{x0}}\delta_{xy}. \tag{7.8}$$

Proof

$$\langle x|D|y\rangle = \langle x|\left[-I + 2|\varphi_o\rangle\langle\varphi_0|\right]|y\rangle = -\delta_{xy} + \frac{2}{N}$$

$$|\varphi_0\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle$$

$$\langle x|W_n R_0 W_n|y\rangle = \sum_{u,v}\langle x|W_n|u\rangle\langle u|R_0|v\rangle\langle v|W_n|y\rangle = \frac{1}{N}\sum_{u,v}(-1)^{x\cdot u}(-1)^{1-\delta_{u0}}\delta_{uv}(-1)^{v\cdot y}.$$

$$= \frac{1}{N}\sum_{u}(-1)^{x\cdot u}(-1)^{y\cdot u}(-1)^{1-\delta_{u0}}$$

$$= \frac{1}{N}\left[1 - \sum_{u\neq 0}(-1)^{x\cdot u}(-1)^{y\cdot u}\right] = A$$

x = y: $$A = \frac{1}{N}\left[1 - \sum_{u\neq 0}\right] = \frac{1}{N}\left[1 - (N-1)\right] = -1 + \frac{2}{N}$$

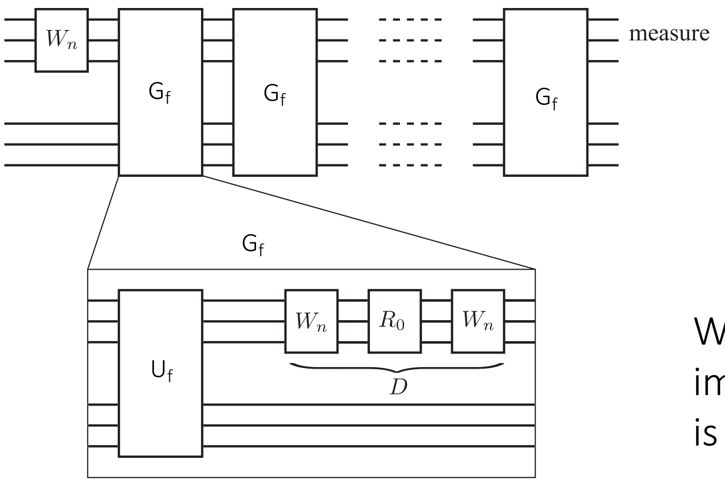
x ≠ y. As discussed in relation to the Deutsch-Jozsa algorithm

$$\sum_{u=0}^{N-1}(-1)^{x\cdot u} = 0 \;\rightarrow\; \sum_{u\neq 0}^{N-1}(-1)^{x\cdot u} = -1$$

Therefore: $A = \frac{1}{N}\left[1 - (-1)\right] = \frac{2}{N}$

Back on Grover's search algorithm

Therefore the D gate can be implemented efficiently. The overall circuit is



We are not interested on how to implement the oracle U_f since this is supposed to be given

Optimality of Grover's algorithm. We have shown that a quantum computer can search N items, consulting the search oracle only $O(\sqrt{N})$ times. One can prove that no quantum algorithm can perform this task using fewer than
$O(\sqrt{N})$ accesses to the search oracle, and thus the algorithm we have demonstrated is optimal.

For a proof, see e.g. Nielsen – Chuang p. 269.