Quantum Algorithms : Grover's Search

Dhiraj Madan *

June 2021

1 Introduction

Consider a large (and unordered) database of size N, consisting of N elements $a_1, ..., a_N$. The problem is to search for a given element y i.e. we need to find an index x, such that $a_x = y$. We assume that the elements are arranged in the form of an unordered array without any index built on top. Classically , one can search for this in linear or $O(N)$ time by iterating through each of the elements one by one. The question is can we do better? Well classically we can not. This is because we will need to look at each element at least once which will consume linear time. However we will see in this lecture that it is indeed possible to obtain a quadratic improvement over this lower bound in quantum setting through the Grover's search algorithm [1] which we will describe next i.e. we will be able to perform the above search using an $O(\sqrt{N})$ depth quantum circuit.

2 Oracle Search problem

Our algorithm is indeed more general and works for general oracle search problems. An **oracle** is essentially a black box function $f : [N] \mapsto \{0, 1\}$. Here we denote by $[N]$ the set of indices $\{1, 2, ..., N\}$. One is only given a query access to the oracle. The problem is to find an x where $f(x) = 1$ in minimum number of queries. Again this takes $O(N)$ queries classically. Note that the problem of database search is also a form of oracle search problem. Here given a search element y, one can define an oracle f_y as follows:-

$$f_y(x) = \begin{cases} 1 & \text{if } a_x = y \\ 0 & \text{Otherwise} \end{cases} \tag{1}$$

Now searching for an element with $f_y(x) = 1$ is equivalent to searching for x such that $a_x = y$. From hereon, we now consider the more general problem of finding x where $f(x) = 1$.

*Advisory Research Engineer, IBM Research India

2.1 The quantum oracle

In quantum setting one has the capacity to use a quantum oracle. This allows one to query the function f in superposition. As in general quantum operations we model this using a reversible unitary mapping defined as follows:-

$$O_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle \tag{2}$$

Here we use 2 quantum registers a query register with $\lceil log(N) \rceil$ qubits for input query $|x\rangle$ and a 1 qubit register $|y\rangle$ to receive the output. Note that if the second register is $|0\rangle$, one indeed gets $f(x)$ in the second register after the application of oracle:-

$$O_f |x\rangle |0\rangle = |x\rangle |0 \oplus f(x)\rangle = |x\rangle |f(x)\rangle \tag{3}$$

Note that if the second register is $|1\rangle$, one indeed gets the flipped value of $f(x)$ i.e. $\overline{f(x)}$ in the second register :-

$$O_f |x\rangle |1\rangle = |x\rangle |1 \oplus f(x)\rangle = |x\rangle |\overline{f(x)}\rangle \tag{4}$$

A magic happens when the second register is indeed in superposed state $|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$:-

$$\begin{aligned}
O_f |x\rangle |-\rangle) &= O_f |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) && (5)\\
&= \frac{O_f |x\rangle |0\rangle - O_f |x\rangle |1\rangle}{\sqrt{2}} && (6)\\
&= \frac{|x\rangle |f(x)\rangle - |x\rangle |\overline{f(x)}\rangle}{\sqrt{2}} && (7)\\
&= \begin{cases} |x\rangle |-\rangle & \text{if } f(x) = 0 \\ -|x\rangle |-\rangle & \text{if } f(x) = 1 \end{cases} && (8)\\
&= (-1)^{f(x)} |x\rangle |-\rangle && (9)
\end{aligned}$$

Thus we are able to convert the oracle in the form of a phase oracle which essentially multiplies a phase of form $(-1)^{f(x)}$.

3 Querying in superposition

We will use the phase oracle in order to speed up our search. We assume that there is exactly 1 marked element a i.e. $f(a) = 1$ and $f(x) = 0 \forall x \neq a$. Note that a quantum oracle can indeed be applied in superposed state which is the source of our speedup. To begin with let's say we have created a superposition of all states i.e. $|U\rangle = \frac{\sum_{x \in [N]} |x\rangle}{\sqrt{N}}$.
Such a state can be easily created if N is a power of 2 i.e. $N = 2^n$ by performing $H^{\otimes n} |0^n\rangle = |U\rangle$. Henceforth we will assume in this lecture that N is indeed a power of 2.

Problem to ponder 1. *What happens when N is not a power of 2. Can you reduce to the case where N is a power of 2 using a simple modification? You might incur a constant multiplicative overhead in size.*

Let's say we now create the above state $|U\rangle$. We use this as the starting state of the algorithm $|\psi_0\rangle$ i.e.

$$\begin{aligned}
|\psi_0\rangle &= \frac{1}{\sqrt{N}}\sum_x |x\rangle & (10)\\
&= \frac{1}{\sqrt{N}}(|a\rangle + \sum_{x\neq a}|x\rangle) & (11)\\
&= \frac{1}{\sqrt{N}}|a\rangle + \sqrt{\frac{N-1}{N}}\left(\frac{\sum_{x\neq a}|a\rangle}{\sqrt{N-1}}\right) & (12)\\
&= \frac{1}{\sqrt{N}}|a\rangle + \sqrt{\frac{N-1}{N}}|e\rangle & (13)
\end{aligned}$$

where in the last equation we have define $|e\rangle = \left(\frac{\sum_{x\neq a}|a\rangle}{\sqrt{N-1}}\right)$.

Thus our state $|\psi\rangle_0$ lies in the plane spanned by $|a\rangle$ and $|e\rangle$. See Figure 1.

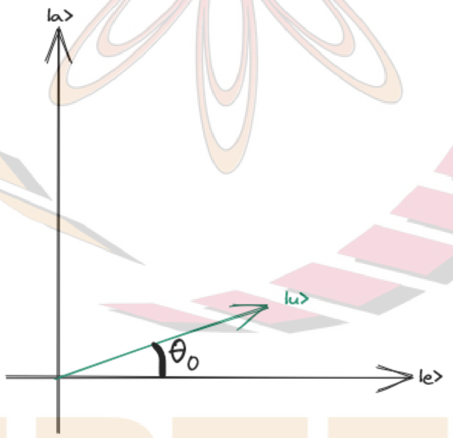


Figure 1: The initial state

We also see that the angle between $|\psi\rangle_0$ and $|e\rangle$, θ_0 is given by :-

$$\cos(\theta_0) = |\langle e|\psi_0\rangle| = \sqrt{\frac{N-1}{N}} \tag{14}$$

or

$$\sin(\theta_0) = \frac{1}{\sqrt{N}} \tag{15}$$

or

$$\theta_0 = \sin^{-1}\frac{1}{\sqrt{N}} \approx \frac{1}{\sqrt{N}} \tag{16}$$

Thus the angle is close 0 for large N, or $|U\rangle$ is quite close to $|e\rangle$. In order to bring it close to $|a\rangle$, we would need a rotation, roughly by $\frac{\pi}{2}$. Is this rotation possible?

3.1 Effect of oracle

First we see how the oracle O_f acts on $|\psi\rangle_0$, since we will need to use this oracle to achieve the desired rotation. We actually consider the more general effect of oracle on vectors in the plane $\text{span}\{|a\rangle, |e\rangle\}$. We observe that :-

$$O_f |a\rangle |-\rangle = (-1)^{f(a)} |a\rangle |-\rangle = -|a\rangle |-\rangle \tag{17}$$

and

$$\begin{aligned} O_f |e\rangle |-\rangle &= O_f \left(\frac{\sum_{x \neq a} |x\rangle}{\sqrt{N-1}} \right) |-\rangle & (18) \\ &= \frac{1}{\sqrt{N-1}} \sum_{x \neq a} O_f |x\rangle |-\rangle & (19) \\ &= \frac{1}{\sqrt{N-1}} \sum_{x \neq a} (-1)^{f(x)} |x\rangle |-\rangle & (20) \\ &= \frac{1}{\sqrt{N-1}} \sum_{x \neq a} |x\rangle |-\rangle & (21) \\ &= |e\rangle |-\rangle & (22) \end{aligned}$$

Thus on a general vector in the plan $\alpha |a\rangle + \beta |e\rangle$, O_f acts as: -

$$O_f(\alpha |a\rangle + \beta |e\rangle) |-\rangle = (-\alpha |a\rangle + \beta |e\rangle) |-\rangle \tag{23}$$

This is equivalent to a reflection about the axis $|e\rangle$. See figure 2 .

4 From reflection to rotation

Essentially we would now need to convert to use the above reflection to perform a rotation. Is this even possible ? Well it is indeed possible if one performs two successive reflections in a plane about different vectors.

Lemma 1. *If $|u\rangle$ and $|v\rangle$ are 2 vectors in a plane making an angle θ. Let $R_{|u\rangle}$ and $R_{|v\rangle}$ be reflections in the plane about $|u\rangle$ and $|v\rangle$ respectively. Then the two successive reflections $R_{|u\rangle} R_{|v\rangle}$ together achieve a rotation in the plane by angle 2θ in the orientation from $|v\rangle$ to $|u\rangle$.*

Proof. The proof is pictorially depicted in Figure 3. Consider any arbitrary vector $|\psi\rangle$ in the plane containing $|u\rangle$ and $|v\rangle$ (Figure 3a). Say it makes an angle ϕ with $|u\rangle$ i.e. an angle of $\theta + \phi$ with $|v\rangle$. Then a reflection about $|v\rangle$ takes it to the opposite side of $|v\rangle$ making an angle of $\theta + \phi$ with $|v\rangle$ (Figure 3b). This

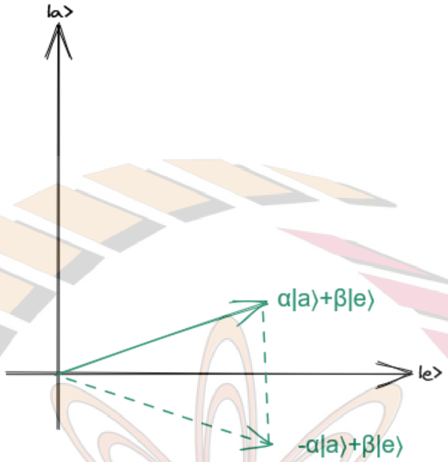


Figure 2: Action of orcale in span$\{|a\rangle, |e\rangle\}$

makes an angle of $2\theta+\phi$ with $|u\rangle$. A further reflection about $|u\rangle$ brings it to the original side of $|u\rangle$ making an angle $2\theta+\phi$ (Figure 3c) . Since the original angle was ϕ the is essentially a rotation by 2θ in the direction from $|v\rangle$ to $|u\rangle$. □

4.1 The second reflection

We are already performing 1 reflection around $|e\rangle$. What is the second reflection axis that can be chosen? Well we can choose this to be the most symmetric vector i.e. $|U\rangle = \frac{1}{\sqrt{N}} \sum_{x\in[N]} |x\rangle$. In this case we know that the angle between the two successive reflection axes is $\theta_0 = \sin^{-1}\frac{1}{\sqrt{N}}$ and hence the two successive reflections will achieve a rotation by angle $2sin^{-1}\frac{1}{\sqrt{N}}$.

The question is how to achieve the above reflection? Firstly we can reduce this to a simpler problem of reflecting about $|0\rangle^n$ using the 3 steps below :-

1. Map $|U\rangle$ to $|0^n\rangle$
2. Reflect about $|0^n\rangle$ i.e. perform $R_{|0^n\rangle}$
3. Map $|0^n\rangle$ back to $|U\rangle$

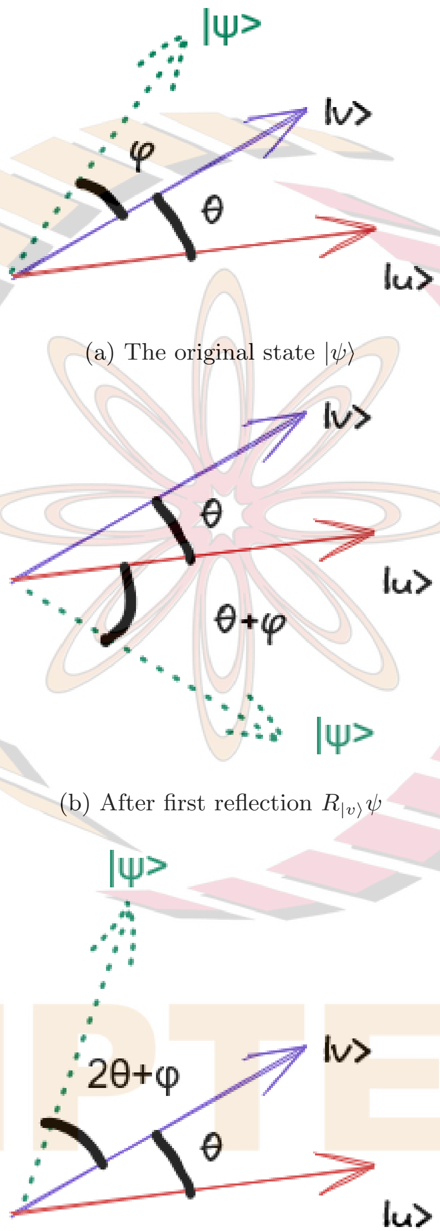
Steps 1 and 2 above can both be achieved using Hadamards since

$$H^{\otimes} |0^n\rangle = |U\rangle \tag{24}$$

and

$$H^{\otimes} |U\rangle = |0^n\rangle \tag{25}$$

.



(a) The original state $|\psi\rangle$

(b) After first reflection $R_{|v\rangle}\psi$

(c) After 2 reflections $R_{|u\rangle}R_{|v\rangle}\psi$

Figure 3: Rotation through two reflections

It now remains to show how to achieve reflection about $|0^n\rangle$ i.e. perform $R_{|0^n\rangle}$. Here we need to be able to perform :-

$$R\,|0^n\rangle = |0^n\rangle \tag{26}$$

and

$$R\,|b_1, ..., b_n\rangle = -\,|b_1, .., b_n\rangle \tag{27}$$

for all other basis elements $b_1, .., b_n$ not equal to 0^n.

This is equivalent to performing $R\,|b_1, .., b_n\rangle = (-1)^{b_1 \vee b_2 ... \vee b_n}\,|b_1, ..., b_n\rangle$. Thus performing an OR operation and translating to a phase oracle gives the above effect.

5 The algorithm and analysis

Thus running the above pair of reflections each time gives us an effect of rotation by angle $2\theta_0$ in the plane. Initially $|\psi\rangle_0$ makes an angle of θ_0 with $|e\rangle$ (See figure 1). Thus after T iterations of the above 2 successive reflections, the angle between the statevector and $|e\rangle$ increases to $(2T + 1)\theta_0$

We need to increase this angle so as to bring our state vector close to $|a\rangle$ or increase the angle close to $\frac{\pi}{2}$. It can be shown that it is always possible to increase the angle to lie in the range $\frac{\pi}{4}$ and $\frac{3\pi}{4}$.

Problem to ponder 2. *Show that it is indeed possible .*

Hint : Make cases on whether θ_0 is $\leq$ or $>$ $\frac{\pi}{4}$. In each case work out if T can be chosen so that $(2T + 1)\theta_0$ lies in range $[\frac{\pi}{4}, \frac{3\pi}{4}]$.

One can now work out that the total number of iterations needed must satisfy :-

$$\frac{\pi}{4} \leq (2T + 1)\theta_0 \leq \frac{3\pi}{4} \tag{28}$$

Thus it suffices to choose

$$2T + 1 = \lceil \frac{\frac{\pi}{4}}{\theta_0} \rceil \tag{29}$$

or,

$$T = \lceil (\frac{\pi}{4\theta_0} - 1)/2 \rceil \tag{30}$$

The above is indeed $O(\sqrt{N})$ for $\theta_0 = sin^{-1} \frac{1}{\sqrt{N}}$.

Thus in $O(\sqrt{N})$ number of reflections one can indeed obtain a state $|\psi\rangle$ which makes an angle less than $\frac{\pi}{4}$ with $|a\rangle$. Now a measurement in standard basis will yield $|a\rangle$ with probability $\geq \cos^2 \frac{\pi}{4} = \frac{1}{2}$.

This gives a randomized algorithm with success probability $\frac{1}{2}$. One can check if algorithm succeeds by computing the oracle f on the measured element.

If it fails the algorithm can be repeated. Note that k independent repetitions of the algorithm will increase the success probability to $\geq 1 - (\frac{1}{2})^k$, bringing it arbitrarily close to 1 with increase in k.

The entire algorithm is summarized in Algorithm 1.

Algorithm 1: Grover's Search Algorithm

```
Create $|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$ using $H^{\otimes n} |0^n\rangle$;
for $t$=1 to $T(= O(\sqrt(N)))$ do
    Reflect $|\psi\rangle_{t-1}$ about $|e\rangle$ using $O_f$;
    Reflect the above about $|U\rangle$ using $H^{\otimes n} R_{|0^n\rangle} H^{\otimes n}$ to get $|\psi_t\rangle$;
end
Measure $|\psi\rangle_T$ in standard basis to get $x$;
if $f(x) = 1$ then
    Return x;
else
    Throw error
end
```

Note that the above bound of $O(\sqrt{N})$ is indeed tight and in fact it is impossible to improve over the same [2]. We end this leaving you with a couple of more problems to ponder on :-

Problem to ponder 3. *What if there are multiple elements marked, say k elements are marked ? How will the algorithm change?*

Problem to ponder 4. *What if the number of marked elements is not known in advance ?*

Hope you enjoyed learning about this algorithm.

References

[1] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

[2] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM journal on Computing*, 26(5):1510–1523, 1997.

[3] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

[4] Abraham Asfaw, Antonio Corcoles, Luciano Bello, Yael Ben-Haim, Mehdi Bozzo-Rey, Sergey Bravyi, Nicholas Bronn, Lauren Capelluto, Almudena Carrera Vazquez, Jack Ceroni, Richard Chen, Albert Frisch, Jay Gambetta, Shelly Garion, Leron Gil, Salvador De La Puente Gonzalez, Francis Harkins, Takashi Imamichi, Hwajung Kang, Amir h. Karamlou, Robert

Loredo, David McKay, Antonio Mezzacapo, Zlatko Minev, Ramis Movassagh, Giacomo Nannicini, Paul Nation, Anna Phan, Marco Pistoia, Arthur Rattew, Joachim Schaefer, Javad Shabani, John Smolin, John Stenger, Kristan Temme, Madeleine Tod, Stephen Wood, and James Wootton. Learn quantum computation using qiskit, 2020.