Grover's Algorithm

Unstructured Search Problem - A Comparison Between Classical and Quantum Methods

Emma Allen

August 2023

Contents

1 Introduction

Quantum computing's origins can be traced back to the mid 20th century when Paul Benioff first proposed a quantum Turing machine, that would exploit quantum mechanics to solve problems that classical computers cannot. Currently our best hope of solving difficult problems is using large supercomputers and high performance computing methods, often powered by thousands of classical CPU and GPU cores based on transistor technology. This technology although powerful is reaching its limit, not to mention the huge computational resources required to run such calculations.

The real world runs on quantum mechanics therefore computers that make calculations exploiting quantum principles represents our best tools for understanding our world. In the late 20th century quantum computers started to appear as useful tools as opposed to theoretical mind experiments with the invention of quantum algorithms, theoretically showing how quantum computers outperform their classical counterpart. Notable examples include Grover's algorithm (1996), a quantum search algorithm that provides polynomial speedup [1] and Shor's algorithm (1994) [2], a factoring algorithm which has the potential to break RSA encryption, potentially breaking a cornerstone of modern day encryption. Even from these few examples, it is clear this emerging technology will have far reaching consequences from cyber-security to finance and physical modelling.

This report will focus on Grover's algorithm and provide an implementation using Python and Qiskit to solve the unstructured search problem and compare with a classical implementation. It will be assumed that the reader is familiar with basic concepts of quantum mechanics but has little knowledge about how these are applied to quantum computing.

2 Background

2.1 Quantum Gates

Thinking about quantum gates requires a shift in perspective—from seeing quantum mechanical events as natural phenomena to viewing them as executable operations in a programmable computer. Quantum gates are unitary operators that act on qubits [3]. These operators are usually described as unitary matrices relative to some orthonormal basis. Unlike classical logic gates, all quantum gates are reversible, leading to interesting consequences.

Common quantum gates include the single qubit Pauli gates, which rotates qubits by 180 deg through specified axis (either x, y, orz. The matrix representation of these are the well known Pauli spin matrices. Quantum gates can have inputs of more than one qubit, for example the CNOT (controlled NOT) requires two qubits. If the qubit is in the is in the $|1\rangle$ state, the CNOT gate negates (flips) the target qubit's amplitudes and if the qubit is in the $|0\rangle$ state, the qubit remains unchanged. This can be encoded into a a matrix:

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{1}$$

More common gates are shown in figure 1

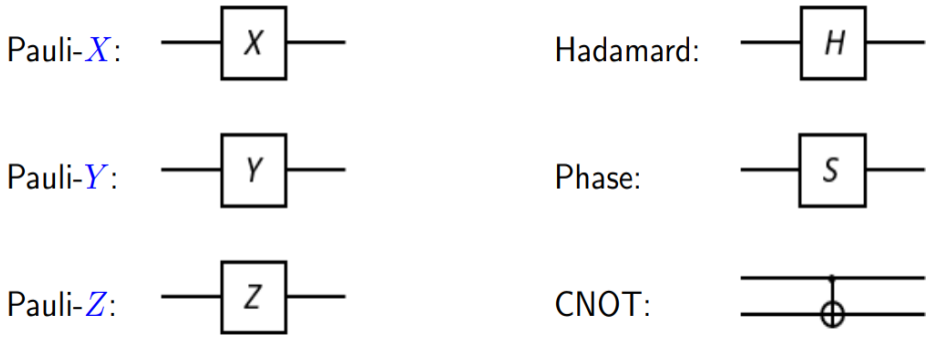

Figure 1: Common Quantum Gates

One more notable single qubit gate is the Hadamard gate, it is one of the most commonly used gates in quantum computing and lacks a classical counterpart. When applied to qubits, it transforms states of 0 or 1 into an equal superposition of both. Its circuit notation can be seen in Figure 1 and as a matrix it is given by:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{2}$$

An interesting property of the Hadamard gate is that squaring it yields the identity matrix. Thus, applying the Hadamard gate twice in a row to the same qubit has no effect.

In summary, quantum gates provide a convenient way to express unitary evolutions, which can also be represented as matrices. Note that this report assumes no noise in the closed quantum system unless explicitly stated.

2.2 Quantum Circuits

Any circuit effectively describes a unitary evolution, in general a quantum circuit has three stages [4]:

- Initialisation: All qubits start in the $|0\rangle$ state.
- Application of quantum gates.
- Final measurements in the computational basis on some or all qubits.

This can be represented in a number of ways, the most common are graphically and as a tensor network, for example the quantum circuit shown in 2, can also be represented as a tensor network as shown:

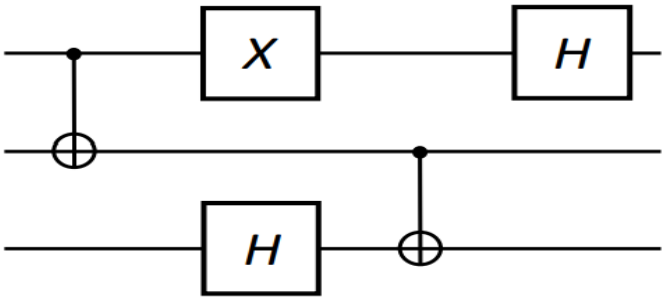

Figure 2: Example Quantum Circuit

$$(H \otimes I_4) \times (I_2 \otimes \text{CNOT}) \times (X \otimes I_2 \otimes H) \times (\text{CNOT} \otimes I_2)$$

where I_2 is the 2x2 identity matrix, and $I_4 = I_2 \otimes I_2$ represents the 4x4 identity matrix.

Composition across wires is achieved via the tensor product, and composition along sets of wires involves matrix multiplication from right to left:

2.3 Bloch Sphere

The Bloch sphere (see figure 3) provides a geometrical representation of the pure state space of a two-level quantum mechanical system (qubit) [5]. The sphere is a mathematical mapping of a complex two-dimensional Hilbert space onto the surface of a unit sphere in three dimensional real space. Every qubit is a combination of the $|0\rangle$ and $|1\rangle$ states, then using polar representation this can be plotted onto a sphere, for further explanation see [6].

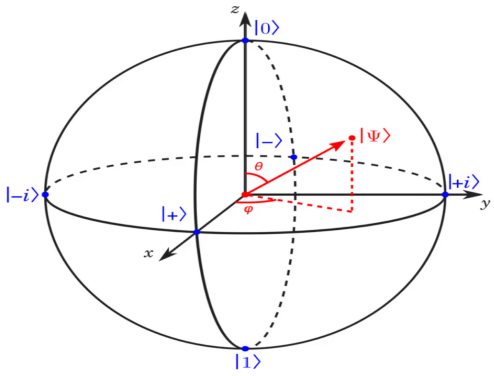

Figure 3: Bloch Sphere

2.4 The Oracle

In classical computing, an oracle is a theoretical black box that answers specific questions about functions, typically returning yes or no. Quantum oracles serve a similar purpose but are implemented as quantum functions. They take advantage of quantum mechanics principles, such as superposition and entanglement. These oracles play a crucial role in quantum algorithms like Grover's and Deutsch-Jozsa and will be used later.

3 Grover's Algorithm

Background / Setting up the problem

Developed in 1996 by Lov Grover, Grover's algorithm is a quantum algorithm that solves the unstructured search problem [1]. Imagine you are given an address book of N names and wish to find one individual in this book. A classical approach would entail to searching through all N elements in order to find the individual, which would take time $O(N)$. Grover's algorithm, on the other hand, can find the marked element in time $O(\sqrt{N})$. This represents a quadratic speed up in comparison to the classical case.

Grover's algorithm essentially allows one to solve the task of function inversion, hence there are uses within cryptography. Grover's algorithm gives broad asymptotic speed-ups to multiple types of brute-force attacks on symmetric-key cryptography, including collision attacks and pre-image attacks.

Mathematical Overview

At a high level, the key elements of Grover's algorithm are [1]:

1. Initialisation to a uniform superposition.
2. The oracle function.
3. Amplitude amplification.

We will consider these in tern, beginning with with initialisation to a uniform superposition, this is done by applying the Hadamard operator to each term (recall the Hadamard operator puts states into a superposition). Suppose we start with the state $|0\rangle^{\otimes N}$ where

$$|0\rangle^{\otimes N} = \underbrace{|0\rangle \otimes |0\rangle \otimes \cdots \otimes |0\rangle}_{N \text{ times}}$$

In which N is the size of the database or list to search through. To each state the Hadamard operator is applied, resulting in $H^{\otimes N}|0\rangle^{\otimes N}$,. This state is a linear combination of all states such that they all have an equal probability of being the target state
To illustrate this further, consider an example with with $N = 3$. The starting state is then $|000\rangle$, and applying the Hadamard operator gives:

$$|\psi\rangle = (H^\dagger \otimes H^\dagger \otimes H^\dagger)|000\rangle = \frac{1}{2\sqrt{2}}(|000\rangle + |001\rangle + |010\rangle + |100\rangle + |011\rangle + |101\rangle + |110\rangle + |111\rangle).$$

This is the end of the initialisation phase.

Next we setup the oracle. In this case the oracle will pick out the target state, for instance the state $|s\rangle$. One applies the oracle to the linear combination state (created in initialisation). Applying the oracle will only flip the sign of the target state (all other states are left unaffected). The oracle function is:

$$\hat{O} = \mathbb{I} - 2|\psi\rangle\langle\psi|$$

where I is the identity operator. This matrix will be diagonal containing ones except for the entry associated with the —s$\rangle$)$which will be negative one(flipping the sign)$.

Back to the example, the state is $|s\rangle = |110\rangle$. then applying the oracle results in the following and marks the end of this phase.

$$|\psi_-\rangle = O|\psi\rangle = \frac{1}{2\sqrt{2}}(|000\rangle + |001\rangle + |010\rangle + |100\rangle + |011\rangle + |101\rangle - |110\rangle + |111\rangle).$$

The final step is amplitude amplification. In this case one applies the Grover operation; A specific case of amplitude application. The Grover operation is defined as:

$$G = 2|\psi\rangle\langle\psi| - \mathbb{I},$$

where all symbols have their previously defined meanings For the example above, this gives:

$$|\psi_d\rangle = \frac{1}{2\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle + |100\rangle + |011\rangle + |101\rangle + 5|110\rangle + |111\rangle).$$

At this point theoretically Grover's algorithm is complete and a measurement will output the probabilities of each state being the target state. however if one iteratively applies G and O $\sqrt{N}$ times, this will cause $|s\rangle$ to have its maximum amplitude and lower the amplitude of the other states. This is called the Grover iteration and doing this makes the results much clearer and is an essential step. Measuring the register after this process is highly likely to yield the state $|s\rangle$.

Further Explanation

To explain this algorithm further, one considers the action of O and G on the desired state $|s\rangle$ and the initial state $|\psi\rangle = H^{\otimes N}|0\rangle^{\otimes N}$:

$$\hat{O}|s\rangle = -|s\rangle$$

$$\hat{O}|\psi\rangle = |\psi\rangle - \frac{2}{\sqrt{N}}|s\rangle$$

$$\hat{G}|s\rangle = \frac{2}{\sqrt{N}}|\psi\rangle - |s\rangle$$

$$\hat{G}|\psi\rangle = |\psi\rangle$$

The action of both operators always returns a linear combination of $|s\rangle$ and $|\psi\rangle$, ensuring that the action of $G \circ O$ on $|\psi\rangle$ remains within the two-dimensional subspace spanned by $|\psi\rangle$ and $|s\rangle$.

The first step of Grover's algorithm is to apply $O^\dagger$ on $|\psi\rangle$. We can decompose $|\psi\rangle$ in terms of $|s\rangle$ and $|s^\perp\rangle$, noting that since each coefficient in $|\psi\rangle$ is equal for a properly normalised state, we must have $\langle s|\psi\rangle = N^{-1/2}$. Therefore, we have:

$$O^\dagger|\psi\rangle = O^\dagger\left(\frac{1}{\sqrt{N}}|s^\perp\rangle + \frac{1}{\sqrt{N}}|s\rangle\right) = \frac{1}{\sqrt{N}}|s^\perp\rangle - \frac{1}{\sqrt{N}}|s\rangle.$$

This can also be considered geometrically as a reflection of $|\psi\rangle$ in the $|s\rangle$ axis (see Figure reflection$_o$*peration*).*If*$|\psi\rangle$ makes an angle θ with $|s^\perp\rangle$, then the action of $O^\dagger$ is a rotation through -2θ.

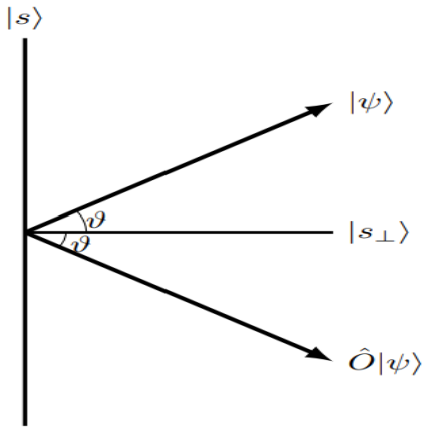

Figure 4: Reflection of $\hat{O}$ on $|\psi\rangle$ along the $|s\rangle$ axis. [7]

Applying $\hat{G}$ can also be interpreted as a reflection in the line defined by the original state $|\psi\rangle$. This can also be viewed as a rotation through 4θ, so $\hat{G}\hat{O}$ has a net result of a rotation of 2θ towards $|s\rangle$.

Each time $\hat{G}\hat{O}$ is applied, it rotates the state 2θ towards $|s\rangle$. At some point, we will go past $|s\rangle$ and start moving away again. Therefore it is essential to know how many times to apply $\hat{G}\hat{O}$ in order to maximise the probability of measuring the register in state $|s\rangle$.

The initial angle θ is related to the original state as shown in Figure [**fig:decomposition˙triangle**]. We have

$$\sin\theta = \frac{1}{\sqrt{N}}.$$

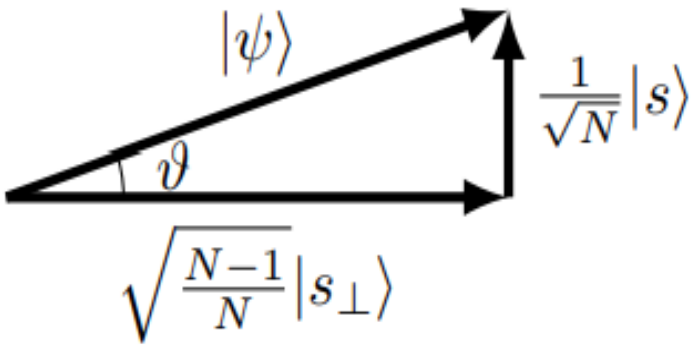

Figure 5: Decomposition of $|\psi\rangle$ into $\alpha|s^\perp\rangle + \beta|s\rangle$. [7]

For large N, θ must be small, so we have $\theta \approx N^{-1/2}$. Similarly, $|\psi\rangle$ is almost entirely aligned with $|s^\perp\rangle$, therefore we need to rotate it through an angle of $\frac{\pi}{2}$ to be aligned in the $|s\rangle$ direction. Since each iteration moves the it an angle $2\theta \approx 2/\sqrt{N}$, we need $\frac{\pi}{4\theta} = \frac{\pi\sqrt{N}}{4}$ (or the closest integer to this) iterations to reach maximal probability of measuring $|s\rangle$, or in other words we need the

nearest integer to $\sqrt{N} iterations$.

Overall, to summarise suppose we have a quantum register $|\phi\rangle$, and we wish to know if a given state $|s\rangle$ is in this register. Then we use the Grover algorithm as described above, and after $\sqrt{N}$ iterations, we measure the register. We expect to measure $|s\rangle$ with high probability if $|s\rangle$ is in $|\phi\rangle$. From this, we conclude that using Grover algorithm to search through N qubits, is $O(\sqrt{N})$, which is provides quadratic speed up over its classical counterpart, requires $O(N)$.

4 Experimental Method

The unstructured search problem is solved using two methods to determine the number of queries needed to find a target number from a list of N integers. Both approaches are implemented in Python. The first method employs a straightforward classical approach, while the second method utilises a quantum simulation with Qiskit version 1.1.1. It is expected that the classical approach requires N searches, while the quantum approach requires approximately $\sqrt{N}$ searches, through this experiment the number of searches is verified.
In this experiment the input is a randomly generated list of integers with length eight; then the target element is randomly selected from this list. The list and target are the same for the classical case and the quantum case. This experiment is run 100 times and then the average number of searches was calculated for each case. The full code can be found at [8].

5 Results and Discussion

5.1 Classical Results

It was found that in 100 trials the average number of searches is 5.88 and the mode index is 7. Although below the expected N average number of calls, this figure is likely within uncertainty ranges and also reflects the random nature of list generation. Statistical analysis would need to be completed to confirm this.

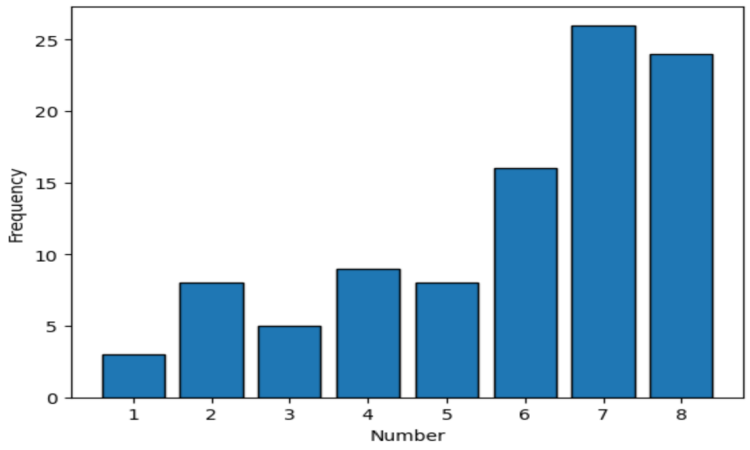

Figure 6: Bar chart displaying the frequency of calls required to find the target element

5.2 Quantum Results

In 100 trials the average number of searches was 1.00. The prediction was $\sqrt{N} = \sqrt{8} \approx 2.83$. Therefore this result is also likely to be within uncertainty ranges and statistical analysis is required to confirm this. The discrepancy in results is likely due to noise. Furthermore there is no error correction in this code which could have contributed to the result not being exactly $sqrt8$. However despite these issues this demonstrates the the number of calls required by Grover's algorithm is nearly six times less than this classical implementation.

Due to the quantum nature of Grover's algorithm, finding the target state is given a probability. Figure 7 displays the probability distribution for one particular trial. One can clearly see the target state has the highest probability of being the target.

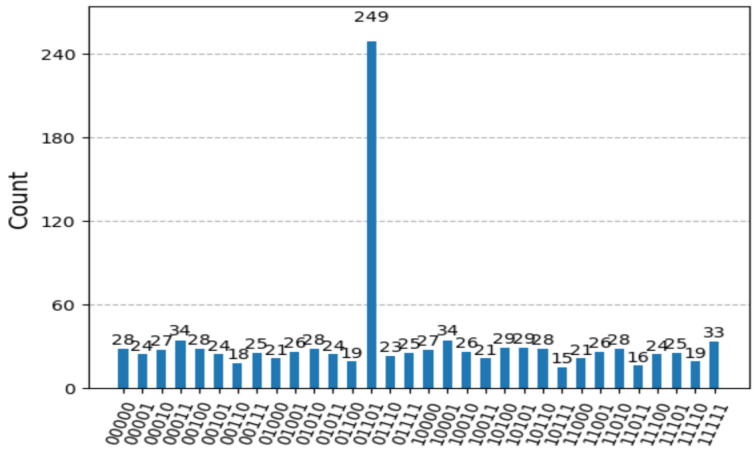

Figure 7: Bar chart displaying number of counts to each state. 10110 has the highest count rate therefore the highest probability of being that target state, which in this trial it was.

6 Conclusions and Future Developments

An overview of Grover's algorithm and its mathematical derivation was given. Then Grover's algorithm was implemented in python and the experiment showed that it outperforms a classical implementation to find a target element within a list of 8 elements.

Furthermore it has been shown that using a quantum simulator and simple classical function that these are likely to agree with their time complexities, although statistical analysis is needed to confirm this. In this experiment in 100 trials on average a classical approach requires 5.88 searches and quantum approach requires 1.00. This numbers are not exactly what is expected, this likely reflects the random nature of the list generation for the classical case and the noise and lack of error correction in the quantum case.

However these results show clearly that quantum approach works far better than the classical case, highlighting the promise of quantum computing in the future.

Given this code was written for a small integer list, In the future to collect more cohesive and convincing results the code should be extended for greater list sizes of 100 or more. At this size error correction would be essential and adjustments should be made for dealing with noise.

Furthermore, this experiment can be extended by searching for other things within a data base such as the maximum number. In addition if the increased list size was successfully implemented, this program could be used to search through a small real-life database.

References

[1] Giacomo Nannicini. *An Introduction to Quantum Computing, Without the Physics*. Last updated: February 24, 2020. Yorktown Heights, NY, 2020.

[2] Lov K. Grover. "A fast quantum mechanical algorithm for database search". In: *Quantum Physics* (1996). Submitted on 29 May 1996 (v1), last revised 19 Nov 1996 (this version, v3). URL: https://arxiv.org/abs/quant-ph/9605043v3.

[3] Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM Review* 41.2 (1999), pp. 303–332. DOI: 10.1137/S0097539795293172.

[4] C.P. Williams. "Quantum Gates". In: *Explorations in Quantum Computing*. Texts in Computer Science. Springer-Verlag London Limited, 2011. Chap. 2. DOI: 10.1007/978-1-84628-887-6_2.

[5] ejallen471. *Unstructred Search - Classical and Quantum*. URL: https://github.com/ejallen471/quantumComputingGrovers.

[6] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010. ISBN: 9781107002173.

[7] Leonardo Castellani. *Lecture Notes on Quantum Computation*. Course material for "Introduction to Quantum Computation" at Laurea Magistrale in Fisica dei Sistemi Complessi, UPO. Alessandria, Italy: Università del Piemonte Orientale, 2021.

[8] Willoughby Seago. *Principles of Quantum Mechanics*. Sept. 2020. URL: https://github.com/WilloughbySeago/Uni-Notes/blob/main/year-3/Principles-of-Quantum-Mechanics/Principles-of-Quantum-Mechanics-Notes.pdf.

[9] Anastasios Kyrillidis. *Quantum Explorations in OptimaLab*. GitHub repository. 2020. URL: https://github.com/akyrillidis/explore-quantum.