IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-5728, p-ISSN: 2319-765X. Volume 19, Issue 2 Ser. II (Mar. – Apr. 2023), PP 55-71
www.iosrjournals.org

Comparing efficiency of the Grover's search algorithm against its classical simulator

Devansh Mishra

Abstract
In recent times, there has been an increasing level of research into quantum algorithms and its uses in order to find more efficient ways of computing. The Grover's Search Algorithm is one such algorithm that has become popular that aims search for a given item in an unsorted database. In the following research experiment, Grover's Algorithm was built using IBM Quantum and its time efficiency was compared against its classical simulator. It was found that the Grover's Algorithm does in fact provide a better time efficiency than its classical simulator, but the correlation between the two times differs from their theoretical prediction.

Date of Submission: 02-04-2023 Date of Acceptance: 13-04-2023

I. Introduction

First discovered in 1996, Grover's algorithm provides a quadratic speedup com-pared to its classical simulator when trying to find a given element in an un-sorted database. If the classical simulator takes O(N) steps to complete the

$$\sqrt{\quad}$$

search, Grover's algorithm would take O(N) steps. [1] The following research attempts to prove this relationship between the time efficiency of the Grover's algorithm on a quantum computer and its classical counterpart on a classical computer.

Cryptography has had vast improvements since the field of quantum computing was introduced. Due to the seminal research conducted by Grover [1] and Shor [2], problems that were unable to be solved using classical methods were now approachable by quantum methods.

Grover's algorithm has applications in cryptography for Quantum Key Dis-tribution (QKD) for sharing keys across a private network. It helps users to identify eavesdropping easily, and thus secures communication channels across a network.[3]

However, Grover's algorithm has shown to be a threat to modern cryptogra-phy as well [4]. Due to the faster speeds of computation, in the case of some encryption algorithms, specifically those that employ modern symmetric ciphers [5], the security measures of communication can be broken down faster by hack-ers, proving this to be a threat.

Thus, the verification of this relation between the Grover's algorithm and its classical simulator is essential to verify its potential benefits and threats to its variety of applications, which is the aim of the following research.

II. Theoretical Background and Mathematical Meth-ods

2.1 How Quantum Mechanics differs from Classical Me-chanics

Classical mechanics usually concerns itself with the study of the macroscopic world and is deterministic in nature. Classical mechanics is applied in many events that we are familiar to in our daily lives. [6]

While some quantum phenomena can be seen in our daily lives as well, quantum mechanics as a discipline largely deals with the microscopic world and is inher-ently probabilistic in nature – rather than providing a deterministic result for a certain phenomenon, it returns a probability distribution of a set of possible outcomes.

The three broad areas in modern physics are quantum physics, which deals with microscopic, subatomic particles like the motion of an electron, relativistic physics, which deals with macroscopic phenomena like planetary motion, and classical physics, which lies between these two extremes. While there is research being conducted in a cross between two of these fields, this classification acts as a broad distinction between the three fields.

The prime example of how the classical mechanics differs from quantum me-chanics is present in the famous Young's double slit experiment [7]. The setup of the experiment is as follows – a gun is placed with a barrier containing two slits in front of it and a backstop that detects the landing of the bullets from the gun behind it.

In the classical sense, the bullets can only go through one of the two slits in the barrier. Thus, the probability of finding it at some location on the backstop is simply the sum of the probabilities from either slit. [8]

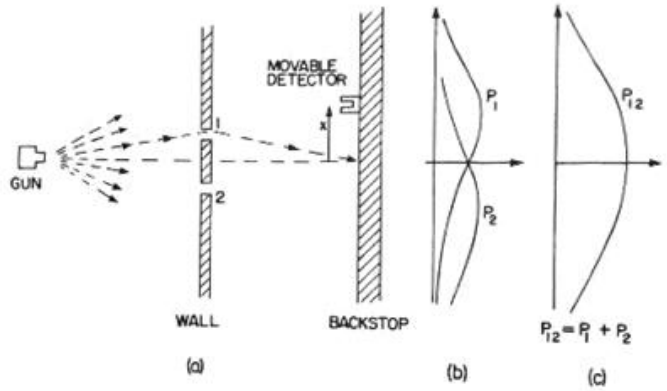

Figure 1: Double Slit Experiment in the Classical World

In the experiment presented in Figure 1, $P_{12} = P_1 + P_2$. P_1 is the probability of the bullet going through the first slit, P_2 is the probability of the bullet going through the second slit, and P_{12} is the combined probability of the two events.

However, in the quantum world, the gun if replaced with an electron gun, then the probability wave of the electron can pass through both slits simultaneously and form an interference pattern on the backstop. This is due to the wave-particle duality of subatomic particles. Thus, the observed probability is not the sum of probabilities, but the sum of probability amplitudes of the two slits. [8]

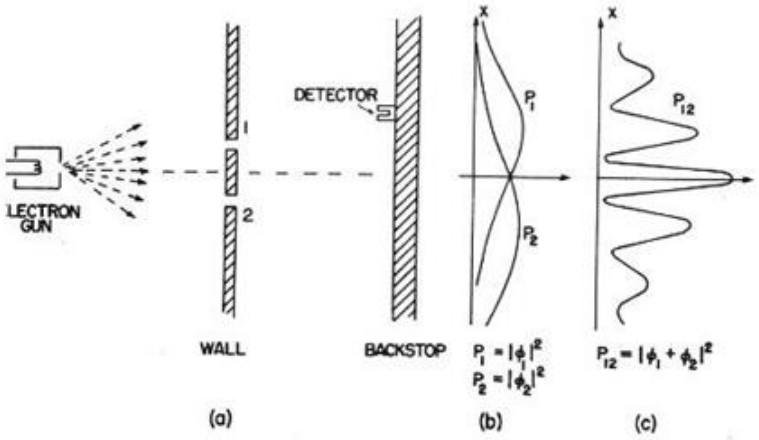

Figure 2: Double Slit Experiment in the Quantum World

In the experiment presented in Figure 2, $P_{12} = |\phi_1 + \phi_2|^2$. $P_1 = |\phi_1|^2$ is the probability of the electron going through the first slit, and $P_2 = |\phi_2|^2$ is the probability of the bullet going through the second slit, and $P_{12} = |\phi_1 + \phi_2|^2$ is the

combined probability of the two events. Here, ϕ represents the probability amplitude of finding a particle in a given position or state. We can expand the result

$$P_{12} = |\phi_1 + \phi_2|^2$$

to give

$$P_{12} = |\phi_1|^2 + |\phi_2|^2 + 2\mathrm{Re}[\phi_1\phi^*_2].$$

Their complex amplitudes can be expressed by their magnitudes and phases to result in

$$P_{12} = P_1 + P_2 + 2\ P_1P_2\cos(\theta_1 - \theta_2), \qquad (1)$$

where θ_1 and θ_2 are the phases of the probability amplitudes ϕ_1 and ϕ_2 respec-tively.

The last term in equation 1 is known as the "interference" term, and it shows that the probability of finding a particle in a given state P_{12} depends on the phase difference between the probabilities P_1 and P_2.

This experiment is the prime example of how quantum mechanics is different from the classical mechanics we are all used to.

2.2 Dirac's notation

To represent quantum systems in an easier way, Paul Dirac introduced his Dirac's notation that used bra-ket notation to represent quantum states. These are vector state symbols that are used to represent quantum systems in differ-ent quantum states. Any ket of the form $|\phi$ can be expressed as the linear combination of basis vectors that span a vector space V as

$$|\phi = \sum_j \phi_j |E_j \qquad (2)$$

where $|E_j$ are all the basisvectors defined on V and any superposition is repre-sented by a unitary linear combination of the two states.

Bras are used to construct maps from a vector space V to its corresponding dual vector state V^*. Any bra $\phi|$ can be constructed by

$$\phi| = |\phi^\dagger = \sum_j \phi^*_j\ E_j| .$$

This allows us to write the scalar product of two vectors as

$$\phi\ \psi = \sum_j \phi^*_j \psi_j$$

and it follows from this that

$$\psi|\psi = |\psi|^2. \qquad (3)$$

The quantity $|\psi|^2$ can be conceptualised as the Length^2 of the vector ψ. [8]

2.3 Qubits

In a classical system, bits are represented by a Boolean value of either 0 or 1, and these are the only two states that the bit can exist in. However, in the quantum world, a quantum bit, or a "qubit", can exist as a superposition of two basis states that exist in a two-dimensional space. Conventionally, qubit basis states are represented by

$$|0 = \begin{matrix}1\end{matrix} \text{and } |1 = \begin{matrix}0\end{matrix},$$

$$0 \qquad\qquad 1$$

and thus, any qubit $|\phi$ can be represented by

$$|\psi = a_0 |0 + a_1 |1 ,$$

where $|0$ and $|1$ are the arbitrary states defined previously, and the values a_0 and a_1 are the normalised coefficients of the vectorϕ along the vectors $|0$ and $|1$ respectively. The probability of finding $|\psi$ in the state $|0$ is $|a_0|^2$ and in the state $|1$ is $|a_1|^2$. By Born's rule [9], we know that a_0 and a_1 cannot be arbitrary complex numbers, but must follow the rule that

$$|a_0|^2 + |a_1|^2 = 1. \qquad (4)$$

All probabilities add up to 1 since it is the sum of the probabilities of the events that could possibly occur in the system – there is nothing else that could happen [11].

2.4 Entanglement

When it is required to represent systems with multiple qubits, tensor products can be used to represent them together [12]. For example, in a two-qubit system, where both qubits are $|0$, the system can be represented as

$$|0 \otimes |0 = |00 .$$

Other states can be represented analogous to this one. Tensor products are also distributive -

$$|0 \otimes (|0 + |1) = |00 + |01 .$$

Tensor products are useful since they provide a mathematical method to repre-sent multi-qubit systems.

Entanglement is a phenomenon that states that two qubits can be intimately linked together, even if they might be far apart from each other in the physical sense. Mathematically, entanglement can be expressed in the following way. If there is a quantum system in the state

$a |00 + b |01 + c |10 + d |11 ,$

and it cannot be broken down in such a way that

$$a |00 + b |01 + c |10 + d |11 = |\psi_1 \ |\psi_2 \ , \qquad (5)$$

then the qubits are said to be in an entangled state. Otherwise, if the system can be broken down in the above way, then it is said to be in a separable state. It follows from this representation and definition that if the coefficients have the property that

$$ad - bc \neq 0, \qquad (6)$$

then the state is entangled; otherwise, it is separable [10].

The physical consequence of an entangled quantum system is that if one of the qubits is measured, then it is possible to know the state of the other qubit without measuring it and disturbing its current state.

2.5 Measurements and Observables

Measurement must be conducted to obtain some result from a quantum op-eration. However, to measure a qubit will disturb the system and will force the qubit to collapse into a single state instead of a superposition of multiple states. This disturbance of the wavefunction is formally called a "collapse" of the wavefunction.

Suppose you have some quantum state $|\psi = a_0 |0 + a_1 |1$, and you measure it with some $|k$ (k=0,1), the output of the measurement will be

$$|a_k|^2 = |\,k|\psi\,|^2$$

$$= \quad \psi|k\ \ k|\psi$$

$$= \psi|\,P_k\,|\psi\,, \qquad (7)$$

where $P_k = |k\ k|$ is a projector on $|k$. This is the collapse of the wavefunction of $|\psi$, and the state is disturbed once it is measured.

An observable is any physical property of a quantum state that can be measured (energy, spin, position, etc.). Any observable A can be represented by

$$A = \quad \lambda_k\,|e_k \quad e_k|$$

$$k$$

$$= \quad \lambda_k P_k, \qquad (8)$$
$$k$$

where P_k is the projector of $|e_k$, and λ_k is its corresponding eigenvalue.

2.6 Quantum Gates and Pauli Matrices

As explained previously, measuring qubits will disturb their original states. However, performing a quantum operation on it will alter the state of the qubit. An operation is not an observation, however, and the qubit can still exist as a nontrivial superposition of states after an operation is performed on. Mathemat-ically, that qubit vector will go through a matrix, giving an output in the same vector space. One of the most fundamental quantum gates is the Hadamard

(H) gate, that is responsible for opening and closing interference patterns. As a matrix, the Hadamard gate is defined as

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

Performing the operation $H\,|0$ will give the result

$$H|0 = \frac{1}{\sqrt{2}}\,(|0 + |1\,),$$

and performing $H\,|1$ will give the result

$$H|1 = \frac{1}{\sqrt{2}}\,(|0 - |1\,).$$

It turns out, however, that in different bases, there are a lot of different op-erations that can be performed, but at the fundamental level, there are four matrices that can perform most operations. These are 1, X, Y, Z - the identity matrix, and three Pauli matrices. Mathematically, these matrices can be represented as

$$1=\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix},$$

$$X = \quad \begin{matrix}0 & 1\\ 1 & 0\end{matrix} \quad ,$$

$$Y = \begin{matrix} 0 & -i \\ i & 0 \end{matrix},$$

$$Z = \begin{matrix} 1 & 0 \\ 0 & -1 \end{matrix}.$$

These matrices each have a special function – 1 is the identity matrix (leaves the qubit unchanged), X is the bit-flip matrix (changes the Boolean value of the bit), Z is the phase-flip matrix (switches the sign of the bit), and Y is the phase-bit-flip matrix (switches both the sign and the bit).All these four matrices are unitary and Hermitian, and a composition of these fundamental building blocks can be used to engineer arbitrary logical operations within a quantum circuit.[11]

2.6.1 Controlled gates

In order to control the operation on a qubit, controlled gates are used to manip-ulate the operation on one qubit depending on the control value from another control qubit. Thus, it is necessary for qubits to have gates that depend on the quantum state of another qubit. These gates are called controlled gates. The most popular controlled gate is the controlled-NOT gate or the CNOT gate. This gate can be decomposed mathematically as [11]

$$c - NOT = |0\ \ 0| \otimes 1 + |1\ \ 1| \otimes X. \tag{9}$$

In matrix form, the c-NOT gate can be represented by

$$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{matrix}.$$

The basic function of the c-NOT gate is that in the incoming input from the two qubits, if the first qubit is in the $|0$ state, then the second qubit is left the same, and if the first qubit is in the $|1$, then the second qubit is bit-flipped.

All controlled gates can be written in the form of c-U, which is the controlled U-gate, where U is an arbitrary single-qubit unitary transformation.

$$c - U = |0\ \ 0| \otimes 1 + |1\ \ 1| \otimes U. \tag{10}$$

These gates allow us to work on multi-qubit systems and manipulate them according to our needs.

2.7 Quantum circuits

Quantum circuits are a compilation of different quantum gates in an order that transforms qubits in the fixed order in which they appear. A quantum program overall can be represented by a sequence of quantum circuits and non-concurrent classical computation.

A typical visual representation of a quantum circuit consists of a series of lines that depict different qubits entering a series of gates and operators and finish with a measurement of the qubits. The most basic quantum circuit that involves one qubit is shown below - [11]

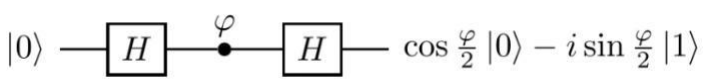

Figure 3: Hadamard-phase circuit

These gates, along with the four basic Hermitian operators of 1, X, Y , and Z, can be combined in such ways in a quantum circuit to perform operations on qubits.

2.8 Grover's algorithm

The primary purpose of Grover's algorithm is to speed up the search of an ele-ment in an unstructured database.

For example, if you are given with a list of N items and you must find the some $x = x_0$ in this list. Classically, the worst case scenario would be looking through the entire list and finding $x = x_0$ as the last item, that is, going through all N items. On an average, the classical algorithm will have to go through N/2 number of steps to find the desired item in the list. [1]

However, the same problem can be solved faster using the Grover's search al-gorithm, which is the quantum mechanical algorithm. Grover's algorithm will $\sqrt{}$ —

take roughly only N number of steps to find the item in an unstructured list, lending a quadratic speedup to the process.

The primary idea of the Grover's algorithm is the usage of an oracle that is designed such that

$$U\,x = \begin{cases} |x, & \text{if } x \neq x_0 \\ -|x, & \text{if } x = x_0, \end{cases} \qquad (11)$$

where |x is an element in the database.

For example, if we have $x_0 = 10$, the oracle matrix can be represented by

$$U_{X_0} = \begin{matrix} 1000 \\ 0 \\ 0 \\ 0 \end{matrix} \begin{matrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{matrix},$$

which essentially "marks" the item corresponding to $x = x_0$. Next, a diffuser circuit is implemented following the oracle that takes the form of

$$G = 2\,|x\;\;x| - 1. \qquad (12)$$

This observable is also called the Grover's observable as it tends to map the marked item to the desired result. At the end of algorithm, the state of the system is altered exactly to $x = x_0$, giving the final result of the search.

III. Methodology

For the following research, quantum and classical systems were required in order to compare the time each of them took to find the same result in an unsorted database.

3.1 Quantum simulator

In order to build the quantum system and its adjoining circuits, the IBM quantum simulator was employed. IBM has employed a collection of high-performance simulators to help its users prototype quantum algorithms under realistic conditions remotely [13]. As available through the IBM Cloud, IBM Quantum offers users with a platform to create and test their quantum circuits and procure data from the same. This is the system that was used for the fol-lowing research in order to obtain the data cited in the following section.

The details of the circuit are present in section 9.2. There are three fundamental sections to each algorithm, as is visually seen - the first is the initialisation of the algorithm (the init); the second is the oracle; and the third is the diffuser that maps the oracle marker onto its corresponding element.

For example, taking the circuit in section 9.2.1 that performs a search for |0010 , the init opens the interference in the circuit. Then, in the following section of the circuit, the oracle is created of the form

1000

$$U_{X_0} = \begin{matrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{matrix} ,$$

such that the element $x_0 = 11$ is marked in the oracle as the target element to find from the search. Then, the diffuser maps this marked oracle onto the element $x_0 = 11$, completing the search.

3.2 Classical simulator

In contrast, to help compare these quantum values against their classical coun-terparts, a Python program was written and run on the online platform Repl.it. As opposed to the quantum circuit built on IBM Quantum, this Python program performed the same search through an unsorted database in a classical fashion, and the time taken for this search to occur was procured and compared against the quantum values. The Python code involved in this classical simulation of the Grover's algorithm is present in section 9.3.

Thus, these two systems were used to procure the data. The algorithm used for the quantum system used 2, 3, and 4 qubits, resulting in databases of sizes 2^2, 2^3, and 2^4 - database sizes of 4, 8, and 16. For the classical algorithm, these sizes were made and the algorithm was run classically. The data obtained was then tabulated and compared.

IV. Results

The raw data collected from the primary experimentation is cited in section 9.1.

The following data is the processed data from that data.

4.1 Processed Data

In the following subsections, the results of the experiments are compared across the classical and quantum data.

4.1.1 4 item search

Item to find	Average time taken / s	
	Classical	Quantum
|00	4.0	2.7
|01	3.7	2.5
|10	3.7	2.7
|11	4.5	2.4

Table 1: Comparison between time taken in classical and quantum cases for a search in a 4 item database

4.1.2 8 item search

Item to find	Average time taken / s	
	Classical	Quantum
|000	6.9	1.9
|001	4.0	3.4
|010	5.5	3.4

|011	4.2	3.2
|100	5.1	3.6
|101	7.3	3.5
|110	4.8	2.7
|111	4.7	1.8

Table 2: Comparison between time taken in classical and quantum cases for a search in an 8 item database

4.1.3 16 item search

Item to find	Average time taken / s	
	Classical	Quantum
|0000	5.4	3.4
|0001	6.4	2.9
|0010	6.8	2.6
|0011	5.8	3.8
|0100	5.7	3.4
|0101	5.1	2.8
|0110	6.9	3.1
|0111	5.6	2.8
|1000	5.2	2.9
|1001	6.0	3.6
|1010	4.5	2.9
|1011	4.9	2.7
|1100	6.0	2.7
|1101	4.6	3.3
|1110	5.7	2.9
|1111	5.8	2.7

Table 3: Comparison between time taken in classical and quantum cases for a search in a 16 item database

4.1.4 Scaling of time taken for quantum and classical algorithms along with size of database

Number of qubits	Average time taken / s	
	Classical	Quantum
2	3.9	2.6
3	5.3	2.9
4	5.7	3.0

Table 4: Comparison between time taken in classical and quantum cases for a search against the size of the database

Figures 4 - 7 present the time taken for the classical and quantum algorithms to find a certain item in an unsorted list. The quantum times are taken as such from the Processed Data; however, the square root of the classical times are taken. This is because the theory predicts that the quantum times will scale with the square root of the classical counterparts, so the square root of the classical times are taken to compare how similar the two time values are. The average time values for the classical and quantum times are graphed at the end of each graph to provide an overall comparison across items.

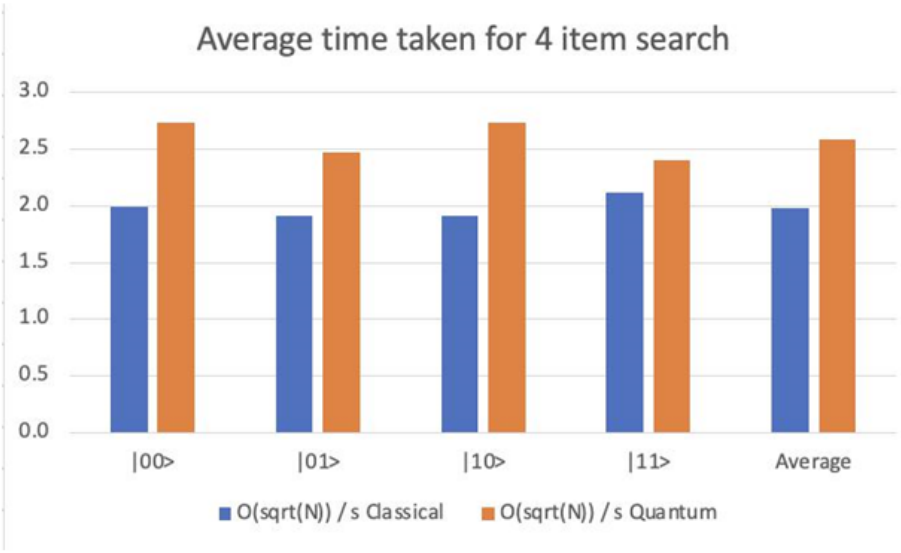

Figure 4: Average time taken for 4 item search

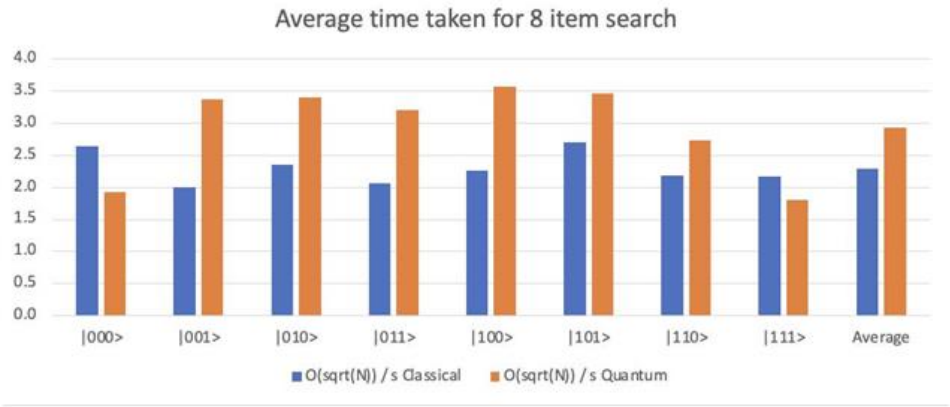

Figure 5: Average time taken for 8 item search

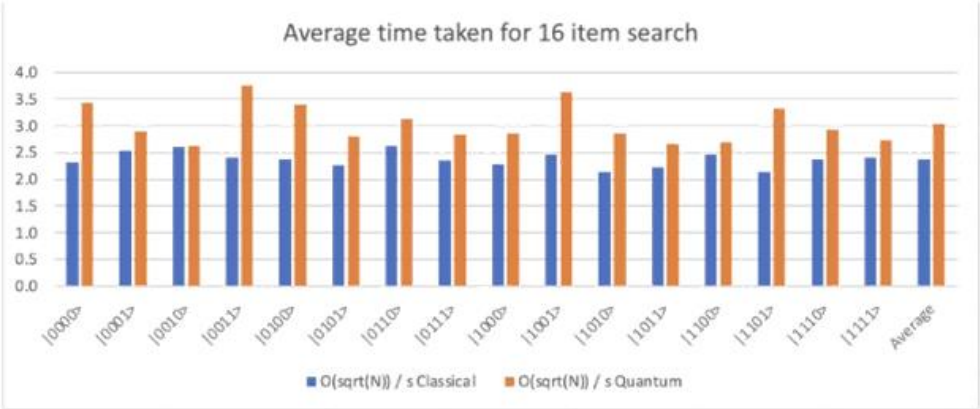

Figure 6: Average time taken for 16 item search

As seen from Figures 4-6, the square root of the average time values for the clas-sical algorithm are roughly around their corresponding quantum times within the range of experimental accuracy.

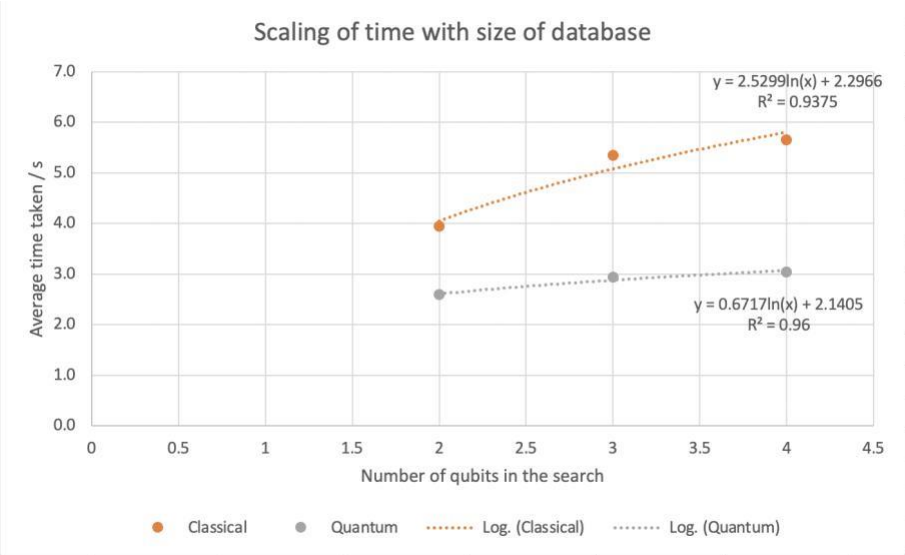

Figure 7: Scaling of time taken for a search through an unstructured database against the size of the database

Figure 7 depicts how the average time values scale with the size of the database. The equations for the trends are -

$$\text{Classicaltrend} \Rightarrow y = 2.5299 \ln(x) + 2.2966 \quad (13)$$

$$\text{Quantumtrend} \Rightarrow y = 0.6717 \ln(x) + 2.1405 \quad (14)$$

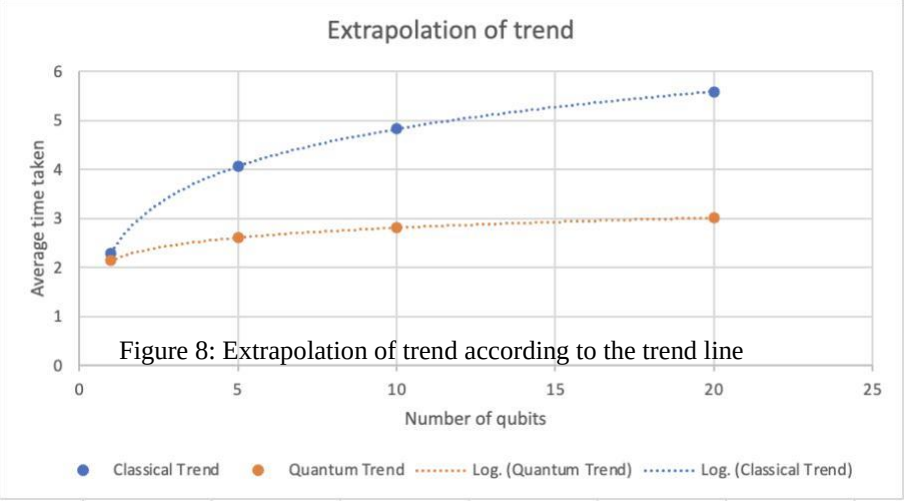

Figure 8: Extrapolation of trend according to the trend line

Figure 8 extrapolates the mathematical trend line observed in 7 across a larger domain and depicts the macroscopic relation between the quantum and classical trends.

V. Analysis

The graphs from the previous section clearly depict the relationship that the classical times and the quantum times along with how they scale with the size of the database they are searching in. However, they appear to be different from the relationship predicted by the theory - both the graphs seem to be following a logarithmic scale from figure 7. This variation from the theoretical prediction could be attributed to the fact that the size of the databases were taken to be smaller than is practically implemented - instead of databases with 4, 8, and 16 items, in practical usage, databases have elements which are of much higher order in magnitude. Thus, at a small scale like the experiment, the average times are seen to deviate from the predicted values.

An interesting aspect to note from figure 8 is that the scaling of the trend lines at a macroscopic level are shown to match the theoretical prediction, how-ever. The quantum trend line is shown to grow at a much slower rate than the classical trend line. This relation can also be depicted using the coefficients of the two trend lines - the quantum trend has a coefficient of 0.6717, while the classical trend line has a coefficient of 2.5299, showing that the quantum trend has a greater time efficiency over its classical counterpart of about 3 times. This depicts the speedup of the Grover's algorithm within the bounds of experimental accuracy [14] [15].

It can also be observed from the bar plots that while there may be differences between the square root of classical times and the quantum times, they average out when looking at the macroscopic trend and follow the predicted trend by becoming roughly equal.

The bar plots, however, depict that the square root classical values are con-sistently lesser than the quantum times. For example, the time comparisons in the 4 item and 16 item search are consistently skewed in this way. This may be attributed to the fact that the algorithm was only run 1024 times for each iter-ation and the fact that the size of the database itself in these values were small. If the algorithm was run for a more number of iterations, then these inconsis-tencies would have become insignificant. Additionally, if the size of databases matched the size that usually found in real-world problems, which are of much larger orders of magnitude, these times would tend to be roughly more or less equal than depicted in figures 4-6.

VI. Evaluation and Limitations

6.1 Strengths

There were strengths involved in the conduction of the above experiment. Firstly, multiple trials were taken across time values for the classical and quantum times to reduce the amount of random error involved in the measurements and make them as precise as possible. This allowed accurate and precise results to be obtained and thus appropriate conclusions could be drawn with ease. Addi-tionally, each trial on the IBM Quantum Composer was taken from a different serve to ensure universality in the measurement of times procured. IBM Quan-tum has open source servers located in locations such as Belgium, Nairobi, and Lima, among others, and taking a variety of these servers for each trial offered universality to the average time value taken. Finally, the R^2 value for both the classical and quantum trends, as depicted in figure 7 were fairly high and close to 1 ($R^2 = 0.9375$ for the classical trend and $R^2 = 0.96$ for the quantum trend), showing that the measurements obtained were accurate and precise in nature.

6.2 Weaknesses

Despite the strengths of the experiment mentioned above, the experimentation had weaknesses involved as well. Firstly, unavailability of a larger and stronger quantum computer made the process of quantum computing slower. This kind of computer was unavailable since larger quantum computers typically require much more physical space than classical computers and are thus not available for open source usage [16]. If there was access to a stronger quantum computer, the quantum algorithm would have been much faster due to the sheer strength that the larger computer provides. Additionally, if the classical algorithm was able to perfectly mimic the procedure of the quantum algorithm, the comparison be-tween the two would have been fairer, making the research outcome closer to the theoretical prediction. However, the classical algorithm was designed in a way that it imitated the function of the quantum algorithm and not the algorithm itself. Finally, it is possible that the observed trend did not match the trend predicted by theory since the search was not conducted at a large enough scale. The theory predicts the Grover's algorithm to provide a quadratic speedup to its classical counterpart in an idealised, large-scale scenario. However, due to the limited scope of this research and the unfeasible nature of building larger databases to search through them, this was unable to be done.

VII. Conclusion

The above research and measurements do not align with the theoretical predic-tion put forth by Grover's search algorithm [1]. This difference can be attributed to the low scale at which the research was conducted - the theoretical relation stems from larger sizes of databases. The measurements obtained in the re-search do not follow the predicted quadratic speedup offered by the Grover's search algorithm. However, the comparison between the quantum and classical trends depict some degree of speedup for the quantum trend, showing that the quantum algorithm is indeed faster than its classical counterpart.

VIII. Appendix

8.1 Raw Data

8.1.1 4 item search

Table 5 shows the results of the time taken from the 4 item search in the IBM Quantum Composer. A circuit was made that created an oracle and helped each item to be found onto a state where it existed on its own.

Item to find	Time taken / s		
	Trial 1	Trial 2	Trial 3
|00	2.7	2.4	3.1
|01	3.3	2.1	2.0
|10	2.6	2.6	3.0
|11	2.4	2.6	2.4

Table 5: Quantum time taken for search in a 4 item database

Table 6 shows the results of the time taken for the execution of the 4 item search in Repl.it. A classical algorithm was made to sort through an unstructured database of 4 items and output the time required to find that item.

Item to find	Time taken / s		
	Trial 1	Trial 2	Trial 3
|00	5.3	2.8	3.8
|01	3.6	3.7	3.7
|10	3.6	3.9	3.5
|11	5.7	3.8	3.9

Table 6: Classical time taken for search in a 4 item database

8.1.2 8 item search

Table 7 shows the results of the time taken from the 8 item search in the IBM Quantum Composer.

Item to find	Time taken / s		
	Trial 1	Trial 2	Trial 3
|000	1.5	1.1	3.2
|001	3.4	3.1	3.6
|010	3.3	3.0	3.9
|011	3.2	2.8	3.6
|100	3.2	4.0	3.5
|101	3.6	3.3	3.5
|110	1.0	3.7	3.5
|111	2.4	2.6	2.4

Table 7: Quantum time taken for search in an 8 item database

Table 8 shows the results of the time taken for the execution of the 8 item search in Repl.it.

Item to find	Time taken / s		
	Trial 1	Trial 2	Trial 3
|000	4.4	4.2	12.2
|001	4.4	3.7	4.0
|010	6.8	6.9	2.9
|011	5.3	3.6	3.8
|100	4.4	7.0	4.0
|101	12.2	5.2	4.6
|110	7.1	3.6	3.7
|111	3.4	5.4	5.3

Table 8: Classical time taken for search in an 8 item database

8.1.3 16 item search

Table 9 shows the results of the time taken from the 16 item search in the IBM Quantum Composer.

Item to find	Time taken / s		
	Trial 1	Trial 2	Trial 3
|0000	2.7	4.6	3.0
|0001	2.8	2.8	3.1
|0010	2.6	2.4	2.9
|0011	2.7	3.0	5.6
|0100	4.6	2.7	2.9
|0101	2.8	2.9	2.7
|0110	2.9	3.1	3.4
|0111	2.8	2.8	2.9
|1000	3.0	2.8	2.8
|1001	3.0	5.0	2.9
|1010	2.9	2.7	3.0
|1011	2.7	2.6	2.7
|1100	2.7	2.7	2.7
|1101	4.6	2.7	2.7
|1110	3.1	2.9	2.8
|1111	2.7	2.7	2.8

Table 9: Quantum time taken for search in a 16 item database

Table 10 shows the results of the time taken for the execution of the 16 item search in Repl.it.

Item to find	Time taken / s		
	Trial 1	Trial 2	Trial 3
|0000	5.6	4.9	5.6
|0001	8.0	5.6	5.6
|0010	5.1	9.4	5.9
|0011	6.9	5.5	5.0
|0100	7.0	5.0	5.0
|0101	5.1	5.9	4.4
|0110	5.9	5.4	9.3
|0111	5.0	5.0	6.7

|1000	5.3	5.7	4.7
|1001	4.7	5.0	8.4
|1010	4.7	4.5	4.4
|1011	4.8	5.0	5.0
|1100	7.8	5.2	5.1
|1101	4.7	4.6	4.5
|1110	6.6	5.0	5.4
|1111	5.7	6.1	5.5

Table 10: Classical time taken for search in a 16 item database

8.2 Quantum circuits for Grover's algorithm

8.2.1 Grover's search for |11

[17]

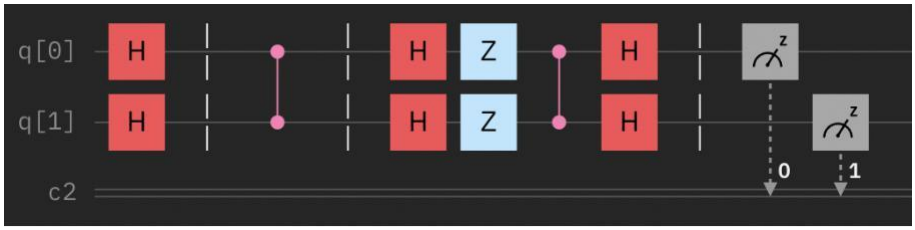

Figure 9: Grover's search for |11

8.2.2 Grover's search for |100

[18]

Figure 10: Grover's search for |100

8.2.3 Grover's search for |0010

[18]

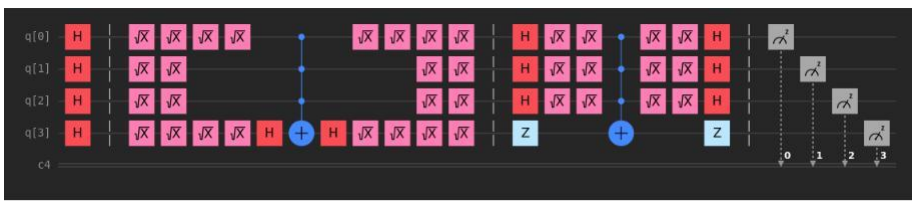

Figure 11: Grover's search for |0010

8.3 Python code for classical simulator of Grover's algo-rithm on Repl.it

```python
import random
import time
def search (ar, num):
    for i in range (1024):
        random.shuffle(ar)
        for i in ar :
            if i == num:
                break
    return 1

def main ():
    st = time.time()
    num = input("What do you want to search for : ")

    if len(num) == 2:
        ar = ['00', '01', '10', '11']
    elif len(num) == 3:
        ar = ['000', '001', '010', '011', '100', '101', '110', '111']
    else:
        ar = ['0000', '00001', '0010', '0011', '0100', '0101', '0110', '0111',
'1000', '1001', '1010', '1011', '1100', '1110', '1111']

    if search(ar, num) == 1 :
        et = time.time()
        elapsed = et-st

    print (elapsed)

main()
```

Figure 12: Python code for classical simulator of Grover's algorithm

The logic of the above algorithm revolves around the search function that takes an unsorted list and a number to find in that list as input. The function then goes through that list and returns the value 1 when the element is found in that list. The main function receives this confirmation and records the time taken for that element to be found from the unsorted list. This forms the essence of the classical simulation of the Grover's algorithm.

References

[1]. Grover, Lov K., "A fast quantum mechanical algorithm for database search", 1996, pp. 212-218
[2]. Shor, P. W. (1994). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, In: Proceedings of the 35th Annual Symposium on the Foundations of Computer Science, edited by S. Goldwasser, p. 124, Los Alamitos, CA, IEEE Computer Society.
[3]. Sakhi, Z. Kabil, R. Tragha, Abderrahim Bennai, Mohamed. (2012). Quantum cryptography based on Grover's algorithm. 33-37. 10.1109/IN-TECH.2012.6457788.
[4]. Mavroeidis, Vasileios et al. "The Impact Of Quantum Computing On Present Cryptography". International Journal Of Advanced Computer Sci-ence And Applications, vol 9, no. 3, 2018. The Science And Information Organization, https://doi.org/10.14569/ijacsa.2018.090354.
[5]. Mina-Zicu, Mina, and Emil Simion. "Threats To Mod-ern Cryptography: Grover'SAlgorithm". Preprints.Org, 2022,
[6]. https://www.preprints.org/manuscript/202009.0677/v1.
[7]. "Difference Between Quantum And Classical Mechanics — Compare The Difference Between Similar Terms". Compare The Difference Between Simi-lar Terms, https://www.differencebetween.com/difference-between-quantum-and-vs-classical-mechanics, 2015
[8]. Wolfgang Rueckner and Joseph Peidle , "Young's double-slit experiment with single photons and quantum eraser", American Journal of Physics 81, 951-958 (2013) https://doi.org/10.1119/1.4819882
[9]. Essler, F.H.L. Lecture Notes On Quantum Mechanics. 2021, pp. 5-6.
[10]. Urbasi Sinha, Christophe Couteau, Zachari Medendorp, Immo S¨ollner, Ray-mond Laflamme, Rafael Sorkin, and Gregor Weihs , "Testing Born's Rule in Quantum Mechanics with a Triple Slit Experiment", AIP Conference Pro-ceedings 1101, 200-207 (2009) https://doi.org/10.1063/1.3109942
[11]. Hosgood, Artur. "Introduction To Quantum Information Science", Problem Sheet. Qubit.Guide, 2022, https://qubit.guide/.
[12]. Ekert, Artur, and Tim Hosgood. Introduction To Quantum Information Science. 2022.
[13]. Quantum Entanglement: A Simple Explanation". Space.Com, 2022, https://www.space.com/31933-quantum-entanglement-action-at-a-distance.html.
[14]. IBM Quantum. https://quantum-computing.ibm.com/, 2021

[15]. "Grover's Search Algorithm — Quantiki". Quantiki.Org, 2022, https://www.quantiki.org/wiki/grovers-search-algorithm
[16]. Xu, Yongzhen et al. "Robust Quantum Walk Search". Arxiv.Org, 2022, https://arxiv.org/abs/2111.09012.
[17]. "Quantum Computers Take Up A Lot Of Space. Re-searchers Decided To Shrink This One Down". ZDNET, 2022, https://www.zdnet.com/article/quantum-computers-take-up-a-lot-of-space-researchers-managed-to-shrink-this-one-down/.
[18]. Norlen, Hassi. Quantum Computing In Practice With Qiskit And IBM Quantum Experience. Packt Publishing, 2020.
[19]. Portugal, Renato. "Basic Quantum Algorithms". Arxiv.Org, 2022, https://arxiv.org/abs/2201.10574.