Journal of Informatics and Web Engineering

Vol. 3 No. 1 (February 2024) eISSN: 2821-370X

Implementation of Grover's Algorithm & Bernstein-Vazirani Algorithm with IBM Qiskit

Yang-Che Liu[1*], Mei-Feng Liu[1]

[1]Xiamen University Malaysia, Malaysia

**Corresponding author: (CST1904731@xmu.edu.my, ORCiD: 0009-0002-6858-5765)*

Abstract - Quantum logic gates differ from classical logic gates as the former involves quantum operators. The conventional gates such as AND, OR, NOT etc., are generally classified as classical gates, however, some of the quantum gates are known as Pauli gates, Toffoli gates and Hadamard gates, respectively. Normally classical states only involve 0 and 1, whereas quantum states involve the superpositions of 0 and 1. Hence, underlying principles of algorithm implementation for classical logic gate and quantum logic gate are indeed different. In this paper, we introduce significant concepts of quantum computations, analyse the discrepancy between classical and quantum gates, compare quantum algorithms using Qiskit against equivalent classical algorithms and analyse their performance in terms of runtime.

Keywords—quantum logic gates, quantum states, quantum computations, quantum algorithms, Qiskit

Received: 21 July 2023; Accepted: 05 September 2023; Published: 16 February 2024

I. INTRODUCTION

With the ever-expanding information being exchanged on the Internet, traditional servers processing these data will be soon hitting an extreme end in the capacity where they be stored. The term "Big Data", referring to large amount of data, requiring adequate storage and processing capability gives rise to huge challenges as researchers try to find the resolution. The needs of high-speed computation and more efficient algorithm had led to the concept of quantum computing [1].

The development of quantum computers has been a popular contemporary topic for decades. In 2019, Google's Sycamore quantum computer equipped a whopping 53 qubits processor [2]; subsequently, researchers from China demonstrated in July 2021 that their quantum computer, Zuchongchi 2.1, can achieve even more qubits than Sycamore. The former is a 56 qubits processor, while Zuchongchi 2.1 is a 66-qubits one [3]. Since a n-qubit quantum computer is equivalent to a 2^n-bit conventional computer, it is expected that quantum computing is a candidate to the solution of high-speed and high-performance algorithm. In comparison, the fastest supercomputer in 2019, namely the IBM's Summit, would be expected to take 10 000 years on a task that Sycamore can achieve in just 200 seconds [2].

A classical bit consists of 0 or 1 state only, while a quantum bit or qubit, can be in the state of 0 and 1 simultaneously, which is known as superposition. In addition, there is a phenomenon called entanglement that in a two-qubit system, if one of the qubit's state is known, the other must also be known. Although there are other approaches to quantum computational logics such as multi-level computational unit called qudit, which provides a larger state space to store and process information, this paper utilizes the conventional 2-level qubit solely [12].

Journal of Informatics and Web Engineering
https://doi.org/10.33093/jiwe.2024.3.1.6

Published by MMU Press. URL: https://journals.mmupress.com/jiwe

One of the objectives of quantum computing is to overcome the limitations of processing power. The vast difference between the operations of classical and quantum bits gives birth to the development of quantum-specific algorithms [5], [11].

There are various fields of quantum computing such as information security which focuses on potential treats to existing cryptography [4]. Also, the cloud quantum computer [6] that provides quantum computing services accessible through internet by the users, such as Qiskit, is utilized in this paper. In addition, physical implementation of quantum computers and logic gates is another popular field of quantum computing [8]. Meanwhile, quantum simulation that emphasizes on modelling the quantum properties of microscopic particles applied in quantum chemistry, material science, and high-energy physics [20], [23] is another example. At the meantime, quantum machine learning is frequently stated as the most promising application for quantum computing [21], [22].

This paper provides a brief introduction to quantum computing, illustrates implementations of quantum algorithms in Qiskit, compares the performance with equivalent classical algorithms, and proposes a Qiskit package for benchmarking quantum algorithms and their classical equivalence.

II. BITS AND QUBITS

A. From Bits to Qubits

Quantum bits can exist as 0 and 1 simultaneously as opposed to classical bits being either only 0 or 1. This phenomenon is known as superposition. The term superposition is known in wave physics as the interference of two or more waves. In quantum physics, when a particle is extremely small, the particle starts to behave like a wave; this is called wave–particle duality, and thus is in a superposition state. However, the difference between the two is that when the quantum particle is observed or measured, this wave collapse, degenerating from the wave into a classical particle. Even up until this moment, physicists still could not understand the mechanism and theory behind this phenomenon, which is referred as the measurement problem of quantum mechanics [16].

B. Dirac Notation

Consider a two 2-dimensional column vectors in Hilbert space with complex entries as shown in Equation (1) where $a, b \in \mathbb{C}^2$

$$a = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}, b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}. \tag{1}$$

The terms $ket\ and\ bra$ are defined in Equation (2), and (3)

$$ket \coloneqq |a\rangle = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}, \tag{2}$$

$$bra \coloneqq \langle b| = |b\rangle^{+} = ({b_1}^{*} \quad {b_2}^{*}), \tag{3}$$

where $|b\rangle^{+}$is the complex conjugate transpose of $|b\rangle$.

Consequently, $bra - ket, ket - bra$, are derived as in Equation (4) and (5)

$$bra - ket \coloneqq \langle b|a\rangle = ({b_1}^{*} \quad {b_2}^{*}) \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = a_1 {b_1}^{*} + a_2 {b_2}^{*} = \langle a|b\rangle^{*} \in \mathbb{C}, \tag{4}$$

Equation (4) is considered as the inner product of $\langle b|$ and $|a\rangle$; thus, a complex number will be produced from the inner product

$$ket - bra \coloneqq |a\rangle\langle b| = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} ({b_1}^{*} \quad {b_2}^{*}) = \begin{pmatrix} a_1 {b_1}^{*} & a_1 {b_2}^{*} \\ a_2 {b_1}^{*} & a_2 {b_2}^{*} \end{pmatrix}, \tag{5}$$

while Equation (5) is considered as the outer product of $|a\rangle$ and $\langle b|$, which produces a 2×2 matrix.

Equation (6) defines two base quantum states

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}, \tag{6}$$

the two states are orthogonal since $\langle 0|1\rangle = 1 * 0 + 0 * 1 = 0$, which will be used to describe and measure quantum states in practice. Although theoretically, there are infinite orthogonal bases, it is simpler to use some common ones, which will be mentioned later in this paper.

For simplicity of calculations, all quantum states are normalised, such that given a quantum state $|\psi\rangle$ defined in Equation (7)

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{7}$$

where $\alpha, \beta \in \mathbb{C}$, it is expected to have the property shown in Equation (8)

$$\langle\psi|\psi\rangle = |\alpha|^2 + |\beta|^2 = 1, \tag{8}$$

since the quantum state $|\psi\rangle$ is described as a linear combination of two base states $|0\rangle, |1\rangle$ with the coefficient of complex numbers α, β where as α^2 is the probability of measuring 0 from the qubit, and β^2 is the probability of measuring 1 from the qubit, the two probabilities sum up to one [17].

C. Measurement

In almost all practical implementations, orthogonal basis is chosen to describe and measure the quantum states. This is known as projective measurements. Although the generalisation of this exists in theory, it is not commonly used in practice, hence the measurement in this study only refer to projective measurement for the sake of simplicity.

Consider a projective measurement of a quantum state $|\psi\rangle$ onto the basis $\{|0\rangle, |1\rangle$. The quantum state $|\psi\rangle$ will collapse into either $|0\rangle$ or $|1\rangle$. This is a z-measurement, since the two states are also the eigenstates of σ_z, which is the Pauli-Z gate that will be discussed later. The eigenstate follows the concept of eigenvector.

As mentioned previously, there are infinitely many different orthogonal bases. Here are some common ones shown in Equation (9), (10), and (11)

$$|0\rangle, |1\rangle, \tag{9}$$

$$|+\rangle \coloneqq \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |-\rangle \coloneqq \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle), \tag{10}$$

$$|+i\rangle \coloneqq \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle), |-i\rangle \coloneqq \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle), \tag{11}$$

where the above bases correspond to the eigenstates of σ_x, and σ_y. Equation (9) is also called as the computational basis, while (10) is known as the Hadamard basis.

Given a quantum state $|\psi\rangle$, the probability of it collapsing to a particular state $|x\rangle$ when performing measurements onto the basis $\{|x\rangle, |x\rangle^+\}$ would be of curiosity. The Born rule defined in Equation (12), is a postulate that enables the probability to be found

$$P(x) = |\langle x|\psi\rangle|^2, \sum_i P(x_i) = 1. \tag{12}$$

As an illustration shown in Equation (13), for a quantum state $|\psi\rangle = \frac{1}{\sqrt{3}}\left(|0\rangle + \sqrt{2}|1\rangle\right)$ measured onto the basis $\{|0\rangle, |1\rangle\}$, the probability of measuring 0 is a third

$$P(0) = \left|\left\langle 0\right|\frac{1}{\sqrt{3}}\left(|0\rangle + \sqrt{2}|1\rangle\right)\rangle\right|^2 = \left|\frac{1}{\sqrt{3}}\langle 0|0\rangle + \frac{2}{\sqrt{3}}\langle 0|1\rangle\right|^2 = \left|\frac{1}{\sqrt{3}} + 0\right|^2 = \frac{1}{3}. \tag{13}$$

Consequently, the probability of measuring 1 will be two thirds, as shown in Equation (14)

$$P(1) = 1 - P(0) = \frac{2}{3}, \tag{14}$$

as another example shown in Equation (15), measuring on a different basis $\{|+\rangle, |-\rangle\}$ from a quantum state $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ results in a certainty for measuring $|+\rangle$

$$P(+) = \left|\left\langle \frac{1}{\sqrt{2}}(\langle 0| + \langle 1|) \middle| \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right\rangle\right|^2 = \left(\frac{1}{4}\right)|\langle 0|0\rangle - \langle 0|1\rangle + \langle 1|0\rangle - \langle 0|0\rangle|^2$$

$$= \left(\frac{1}{4}\right)|1 - 0 + 0 - 1|^2 = 0, \tag{15}$$

consequently, the probability of measure $|-\rangle$ is shown in Equation (16)

$$P(-) = 1, \tag{16}$$

this is expected, as $\langle +|\psi\rangle = \langle +|-\rangle = 0$.

The Born rule has not been proven to be mathematically correct. However, no quantum experiments showed otherwise. Scientists have been following this postulate, whereas physicists cracked their heads to prove Born rule in a mathematical perspective [18].

D. Bloch Sphere

Quantum computing is closely related to linear algebra, in which they can be illustrated visually using a Bloch Sphere. Consider a normalized pure state specified in Equation (17)

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle, \tag{17}$$

where $\varphi \in [0, 2\pi)$ is used to describe the relative phase and $\theta \in [0, \pi]$ is used to derive the probability of measuring $|0\rangle$ or $|1\rangle$, for example $P(0) = \cos^2\frac{\theta}{2}$, $P(1) = \sin^2\frac{\theta}{2}$.

All normalized pure state defined above can be visualised using a Bloch Sphere. They sit on the surface of the sphere with radius 1. Consequently, the mixed states reside within the Bloch Sphere. To illustrate the pure state using the Bloch Sphere, a Bloch vector defined in Equation (18) is used as a representation

$$\vec{r} = \begin{pmatrix} \sin\theta\cos\varphi \\ \sin\theta\sin\varphi \\ \cos\theta \end{pmatrix}, \tag{18}$$

this is the general representation of the sphere coordinate system.

Common orthogonal bases in Equation (19)~(24) can thus be illustrated by the Bloch Sphere in Figure 1 to 3

$$|0\rangle: \theta = 0, \text{arbitary } \varphi, \vec{r} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \tag{19}$$

$$|1\rangle: \theta = \pi, \text{arbitary } \varphi, \vec{r} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}. \tag{20}$$

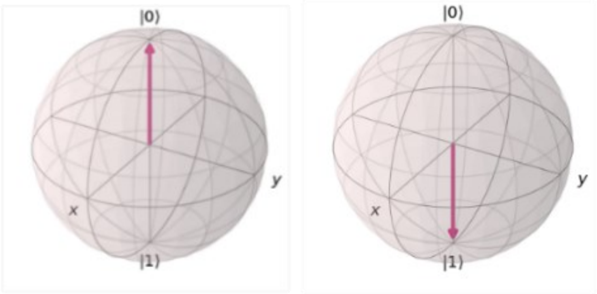


Figure 1. Visualization of $\{|0\rangle, |1\rangle\}$ on Bloch Sphere.

$$|+\rangle: \theta = \frac{\pi}{2}, \varphi = 0, \vec{r} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \tag{21}$$

$$|-\rangle: \theta = \frac{\pi}{2}, \varphi = \pi, \vec{r} = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}. \tag{22}$$

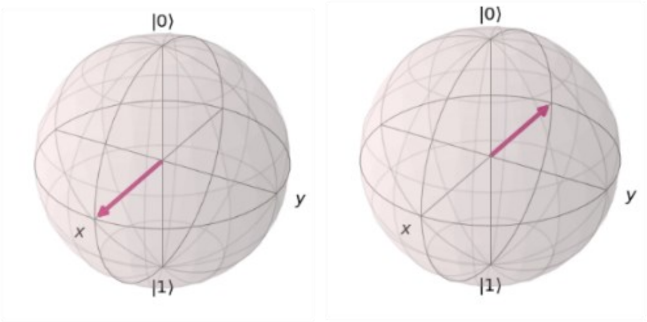


Figure 2. Visualization of $\{|+\rangle, |-\rangle\}$ on Bloch Sphere.

$$|+i\rangle: \theta = \frac{\pi}{2}, \varphi = \frac{\pi}{2}, \vec{r} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \tag{23}$$

$$|-i\rangle: \theta = \frac{\pi}{2}, \varphi = \frac{3\pi}{2}, \vec{r} = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}. \tag{24}$$

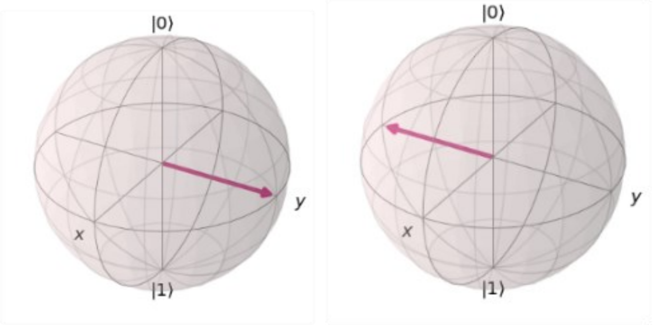


Figure 3. Visualization of $\{|+i\rangle, |-i\rangle\}$ on Bloch Sphere.

Orthogonal bases have angles that are 90° apart from each other. Despite the fact, the above illustration described them with a 180° angle difference. This is a characteristic of the Hilbert space, where orthogonality could be defined.

This results in the normalised pure state having a division of 2 for the angle theta, as shown in Equation (25)

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle, \tag{25}$$

where θ is the angle in the Bloch Sphere and $\theta/2$ is the real angle in Hilbert space.

A z-measurement of a quantum state corresponds to a projection onto the z-axis on the Bloch sphere. If the state is closer to $|0\rangle$, it is more likely to collapse to a $|0\rangle$ state. This is applicable to both x and y-measurements [13].

III. QUANTUM CIRCUITS

A. Basic Single-qubit Gates

Unitary matrices are used to represent various quantum gates in a mathematical perspective. A unitary matrix will have the following characteristic specified in Equation (26),

$$u^{+}u = \mathrm{I}_2, \tag{26}$$

where u is the unitary matrix, u^{+} is the Hermitian conjugate transform of u, and I_2 is the 2×2 identity matrix. This indicates that unitary matrix is reversable [10]. Single qubit gates such as Pauli gate and the Hadamard gate heavily depend upon unitary matrices.

There are three Pauli gates, namely Pauli X, Pauli Y and Pauli Z. They are simply known as the X-gate, Y-gate and Z-gate respectively. A Pauli X gate is defined in Equation (27)

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = |0\rangle\langle 1| + |1\rangle\langle 0|, \tag{27}$$

suppose a Pauli X gate is applied to the quantum state $|0\rangle$, Equation (28) illustrates the representation

$$\sigma_x|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 * 1 + 1 * 0 \\ 1 * 1 + 0 * 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle, \tag{28}$$

similarly, if the Pauli x gate is applied to the quantum state $|1\rangle$, this results in Equation (29)

$$\sigma_x|1\rangle = (|0\rangle\langle 1| + |1\rangle\langle 0|) * |1\rangle = |0\rangle(\langle 1|1\rangle) + |1\rangle(\langle 1|0\rangle) = |0\rangle(1) + |1\rangle(0) = |0\rangle. \tag{29}$$

Equation (28) is described using the matrix notation, whereas (29) takes the Dirac notation. They illustrate that if the initial state is $|0\rangle$, the result is $|1\rangle$ and vice versa. This exhibits the characteristic of a NOT gate, as the X-gate performs an equivalent bit flip operation. In Bloch sphere, this is a rotation around the x-axis by π.

A Pauli z gate is defined in Equation (30) as follows.

$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1|, \tag{30}$$

if the Pauli z gate is applied to the quantum state $|+\rangle$ as displayed in Equation (31)

$$\sigma_z|+\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} * \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix} = |-\rangle, \tag{31}$$

if the Pauli z gate is applied to the quantum state $|-\rangle$ as displayed in Eqution (32)

$$\sigma_z|-\rangle = (|0\rangle\langle 0| - |1\rangle\langle 1|) * \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle, \tag{32}$$

the Z-gate performed a phase flip of the quantum state, this is a rotation around the z-axis by π.

A Pauli y gate is defined in Equation (33)

$$\sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = \mathrm{i} * \sigma_x * \sigma_z, \tag{33}$$

this is a bit-and-phase flip on the input quantum state, and a rotation around the y-axis by π. Since all quantum gates are unitary, it is expected that by applying the Pauli gate twice on the quantum state, the output is the same as the input.

Another frequently used single-qubit gate is the Hadamard gate, defined in Equation (34).

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1|), \tag{34}$$

if the Hadamard gate is applied to the quantum state $|0\rangle$ as presented in Equation (35)

$$H|0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} * \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = |+\rangle, \tag{35}$$

if the Hadamard gate is applied to the quantum state $|1\rangle$ as presented in Equation (36)

$$H|1\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} * \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix} = |-\rangle. \tag{36}$$

The property of changing between x and z made Hadamard gates very frequently used in quantum circuit designs [14]. Qubits are often initialised at $|0\rangle$ state. Applying a Hadamard gate results in a superposition state.

B. Multipartite Quantum States

In practice, many qubits will be manipulated. A tensor product describes a quantum state with multiple qubits, which is known as multipartite quantum states.

Given a qubit A with quantum state $|1\rangle_A$ and a qubit B with quantum state $|0\rangle_B$, the total state, in this case bipartite state is calculated in Equation (37)

$$|10\rangle_{AB} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 * 1 \\ 0 * 0 \\ 1 * 1 \\ 1 * 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \tag{37}$$

Equation (37) depicts that 2 qubits generate a state vector of 4 entries, 3 qubits generate 8 entries, and *n* qubits generate 2^n entries. The property of 2^n results in an exponential increase in speed as compared to classical computers [14].

Some states cannot be written in this form. They are known as correlated states, and sometimes entangled when the correlation between two qubit is strong

$$|\varphi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \tag{38}$$

a correlated state is shown in Equation (38), known as a Bell state. It is used in quantum algorithms which will be discussed later.

C. Basic Two-qubit Gates

An example of two-qubit gates is Controlled NOT (CNOT) gate. The equivalent classical gate, XOR, takes two inputs and generate one output, whereas CNOT gate takes 2 inputs and gives 2 outputs. The latter is unitary, hence reversible.

A CNOT gate is defined in Equation (39)

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = |00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 11| + |11\rangle\langle 10|, \tag{39}$$

if a CNOT gate is applied on a bipartite quantum state $|00\rangle_{xy}$ as demonstrated in Equation (40)

$$CNOT|00\rangle_{xy} = CNOT * \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = |00\rangle_{xy}, \tag{40}$$

if a CNOT gate is applied on a bipartite quantum state $|10\rangle_{xy}$ as demonstrated in Equation (41)

$$CNOT|10\rangle_{xy} = CNOT * \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |11\rangle_{xy}. \tag{41}$$

Table 1 illustrates the truth table for a CNOT gate. When the first qubit is 1, CNOT gate will perform a bit flip on the second qubit. This gives the gate the name Controlled NOT and is also considered a reversible XOR.

Table 1: Truth table for CNOT gate

Input		**Output**	
$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{x}$	$\boldsymbol{x \oplus y}$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

From the CNOT example, it is evident that quantum gates are reversible. Quantum circuits are able to perform any classical functions, and along with the addition of reversible characteristic, it can bring reversibility for the irreversible classical functions [14].

IV. ENTANGLEMENT

An important characteristic of quantum computing is entanglement [9], which is defined as a pure state $|\varphi\rangle_{AB}$ on systems A and B that cannot be expressed as a tensor product of two state, $|\varphi\rangle_A \otimes |\phi\rangle_B$.

A. Bell States

Bell States are maximally entangled and form on orthogonal basis, the term maximally entangled describe the level or degree of entanglement between the two quantum states is at its maximum. The four Bell States are defined as in Equation (42) to (45)

$$|\varphi^{00}\rangle \coloneqq \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \tag{42}$$

$$|\varphi^{01}\rangle \coloneqq \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \tag{43}$$

$$|\varphi^{10}\rangle \coloneqq \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle), \tag{44}$$

$$|\varphi^{11}\rangle \coloneqq \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \tag{45}$$

When a measurement of one qubit is made on a Bell state, the state of the second qubit will be determined without measuring it. Given a Bell state $|\varphi^{00}\rangle$, if the first qubit is measured to state 0, the second must be 0. This phenomenon is known as entanglement. This is widely used in practical quantum circuit design. Equation (46) illustrates a general expression of Bell States [15].

$$|\varphi^{ij}\rangle = (I_2 \otimes {\sigma_x}^{j}{\sigma_z}^{i})|\varphi^{00}\rangle. \tag{46}$$

B. Creating Bell States

Figure 4 shows a circuit of Bell State, where an Hadamard gate is only applied to the upper qubit i_0. The encircled plus representation is a CNOT gate. The CNOT gate flips i_0, known as the target qubit if and only if the j_0 is $|1\rangle$, known as the control qubit.

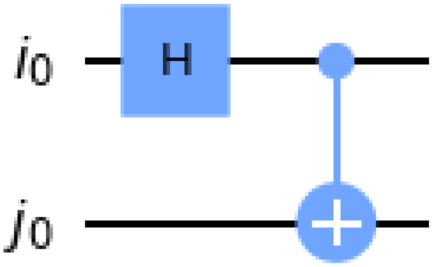


Figure 4. Circuit to create a Bell State.

The Bell measurement determines which of the four Bell states that the two input qubits are in. The circuit is constructed by putting CNOT gate first on one of the qubits (control), and then the Hadamard gate after (target) [15]. Table 2 illustrates the truth table of the circuit in Figure 4.

Table 2: Truth table for Figure 4.1

Initial State $\boldsymbol{|ij\rangle}$	$\boldsymbol{(H \otimes I_2)|ij\rangle}$	$\boldsymbol{CNOT(H \otimes I_2)|ij\rangle}$
$|00\rangle$	$\frac{|00\rangle + |10\rangle}{\sqrt{2}}$	$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$
$|01\rangle$	$\frac{|01\rangle + |11\rangle}{\sqrt{2}}$	$\frac{|01\rangle + |10\rangle}{\sqrt{2}}$
$|10\rangle$	$\frac{|00\rangle - |10\rangle}{\sqrt{2}}$	$\frac{|00\rangle - |11\rangle}{\sqrt{2}}$
$|11\rangle$	$\frac{|01\rangle - |11\rangle}{\sqrt{2}}$	$\frac{|01\rangle - |10\rangle}{\sqrt{2}}$

C. Q-sphere

Bloch Sphere allow visualisation of only a single qubit state. For multiple qubits, Q-sphere is used instead.

The top or “North pole” of the Q-sphere represents the state where all qubits are in state 0, whereas the bottom or “South pole” of the Q-sphere represents the state where all qubits are in state 1. If there are n qubits in the system, thus 2^n basis states, the basis states will be equally distributed as points on the Q-sphere. This is shown in Figure 5.

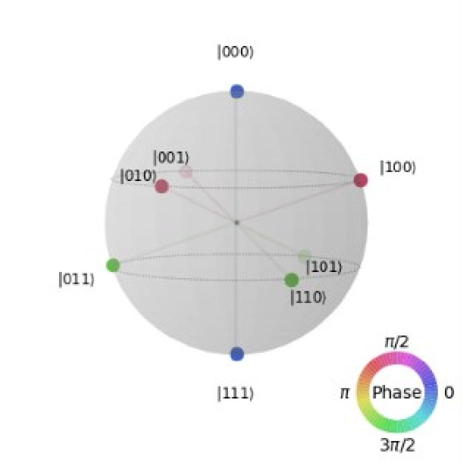
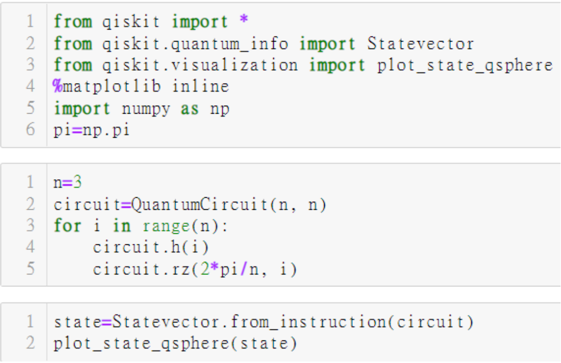


Figure 5. Visualization of a Q-sphere.

The size of the blob (small sphere at each end) is used to represent the probability of measuring the particular basis state, while the colour of the blob is used to represent the phase of the particular basis state. By applying a Hadamard gate for each qubit, we will get evenly distributed state, and by applying rotation around the z-axis for each qubit, the phase will change, and resulting in a difference in the blob’s colour on each set of lines [15].

In Figure 6, a Hadamard gate is applied to every qubit using circuit.h, and circuit.rz denotes a rotation around the z-axis thus altering the phase for each state, while the angle of rotation is $\frac{2\pi}{2}$.



```
from qiskit import *
from qiskit.quantum_info import Statevector
from qiskit.visualization import plot_state_qsphere
%matplotlib inline
import numpy as np
pi=np.pi
```

```
n=3
circuit=QuantumCircuit(n, n)
for i in range(n):
    circuit.h(i)
    circuit.rz(2*pi/n, i)
```

```
state=Statevector.from_instruction(circuit)
plot_state_qsphere(state)
```

Figure 6. Python code to plot a Q-sphere.

V. ALGORITHMS

A. Phase Kickback

Observe the effects of CNOT gate applied on a two-qubit system where the qubits are in superposition state.

$$CNOT|++\rangle = \frac{1}{2}CNOT(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \frac{1}{2}(|00\rangle + |01\rangle + |11\rangle + |10\rangle) = |++\rangle, \quad (47)$$

$$CNOT|+-\rangle = \frac{1}{2}CNOT(|00\rangle - |01\rangle + |10\rangle - |11\rangle) = \frac{1}{2}(|00\rangle - |01\rangle + |11\rangle - |10\rangle) = |--\rangle, \quad (48)$$

$$CNOT|-+\rangle = \frac{1}{2}CNOT(|00\rangle + |01\rangle - |10\rangle - |11\rangle) = \frac{1}{2}(|00\rangle + |01\rangle - |11\rangle - |10\rangle) = |-+\rangle, \tag{49}$$

$$CNOT|--\rangle = \frac{1}{2}CNOT(|00\rangle - |01\rangle - |10\rangle + |11\rangle) = \frac{1}{2}(|00\rangle - |01\rangle - |11\rangle + |10\rangle) = |+-\rangle, \tag{50}$$

which can be further generalized in Equation (51)

$$CNOT|x-\rangle = (-1)^x|x-\rangle, \tag{51}$$

this effect is interesting, since it affects the control qubit rather than the target qubit.

This property is extremely powerful in practical applications since classical oracles can be converted to their quantum equivalent version via this method.

Since the property of CNOT gate shown in Equation (52)

$$|x-\rangle \xrightarrow{CNOT} (-1)^x|x-\rangle, \tag{52}$$

Hence the effect of function f of the system can be derived in Equation (53)

$$|x-\rangle \xrightarrow{U_f} (-1)^{f(x)}|x-\rangle, \tag{53}$$

where U_f denotes the unitary matrix of the function f, illustrated in Figure 7.

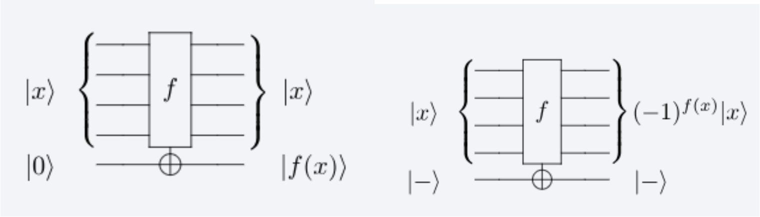


Figure 7. Conversion of classical function to its quantum equivalence.

B. Grover's Algorithm

Grover's algorithm is used for search of a key in databases. In classical algorithms, the algorithm requires O(n) to search through a database to find a record. Lov Grover, in 1996, constructed an algorithm that uses only O($\sqrt{n}$) evaluations, which is a quadratic speedup.

Given a databased of $N = 2^n$ elements, where $n \in \mathbb{N}$, the goal is to find a single element ω. In Grover's algorithm, there are three main steps, namely, state preparation, Grover's oracle, and diffusion operator. The algorithm works by constantly increasing the probability of measuring ω, while shrinking the others via the amplitude amplification procedure.

Figure 8 illustrates the first step, state preparation, the probability of measuring each element is the same. Hence a uniform superposition $|s\rangle$, is constructed by applying Hadamard gate to all qubits as shown in Equation (54)

$$|s\rangle = H^{\otimes n}\,|0\rangle^0. \tag{54}$$

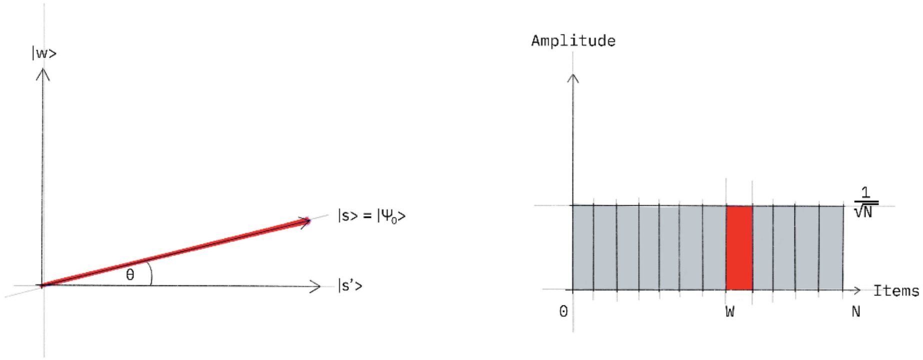


Figure 8. Visualization of step 1.

Figure 9 illustrates the second step, the oracle for any input state $|x\rangle$ in the computational basis is defined in Equation (55)

$$U_\omega|x\rangle = \begin{cases} |x\rangle & if\ x\ \neq \omega \\ -|x\rangle & if\ x = \omega' \end{cases} \tag{55}$$

which can be generalized in Equation (56) using the phase kickback trick

$$U_\omega|x\rangle = (-1)^{f(x)}|x\ -\rangle, \tag{56}$$

a negative phase is added to the solution state so that it can stand out from the rest and be measured.

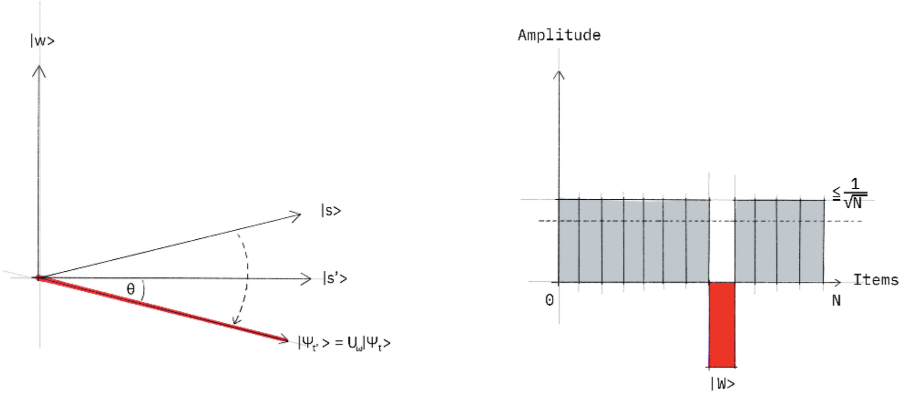


Figure 9. Visualization of step 2.

The oracle is a diagonal matrix, where the corresponding entry of the solution state will have a negative phase. Consider a two-qubit system, and $\omega = 10$, the oracle is constructed in Equation (57)

$$U_\omega = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{57}$$

Finally, figure 10 illustrates the diffusion operator, an additional reflection $U_s = 2|s\rangle\langle s| - I$. Consequently, $|s\rangle$ will be mapped to $U_\omega U_s|s\rangle$. Geometrically, this operation can be interpretated as a reflection about state $|s\rangle$.

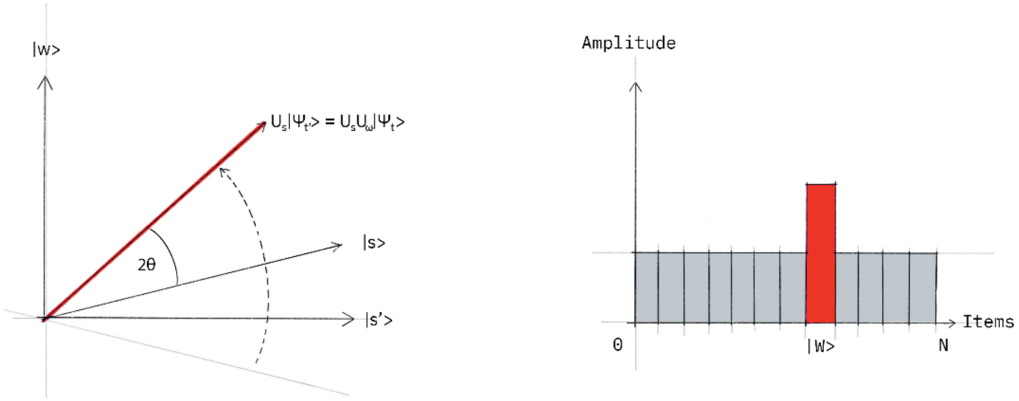


Figure 10. Visualization of step 3.

By repeating step 2 and 3, state $|s\rangle$ will be rotated toward the target element $|\omega\rangle$, and after sufficient amount of iteration, which proves to be $\sqrt{\frac{N}{M}}$, where M is the number of elements that are being found, it is guaranteed to measure the target element $|\omega\rangle$.

C. Bernstein-Vazirani algorithm

Deutsch-Jouza algorithm illustrates two classes of functions categorised as balanced or constant. Bernstein-Vazirani algorithm implements upon this algorithm but attempts to find an encoded string in a function.

Within the Oracle, the function f takes in a string x and outputs either 0 or 1. There is a hidden string s to be found. The function is illustrated in Equation (58):

$$f(x) = s \cdot x \bmod 2. \tag{58}$$

where $s \cdot x = s_0x_0 \oplus s_1x_1 \oplus \ldots \oplus s_{n-1}x_{n-1}$ denotes the sum of the bitwise product.

In the classical approach, n strings will be enquired to the oracle, where each enquires will have a different position of 1, after n attempts, the hidden string can be recovered. As an illustration, s is 101001, the hidden string can be recovered after 6 tries, as shown in Figure 11.

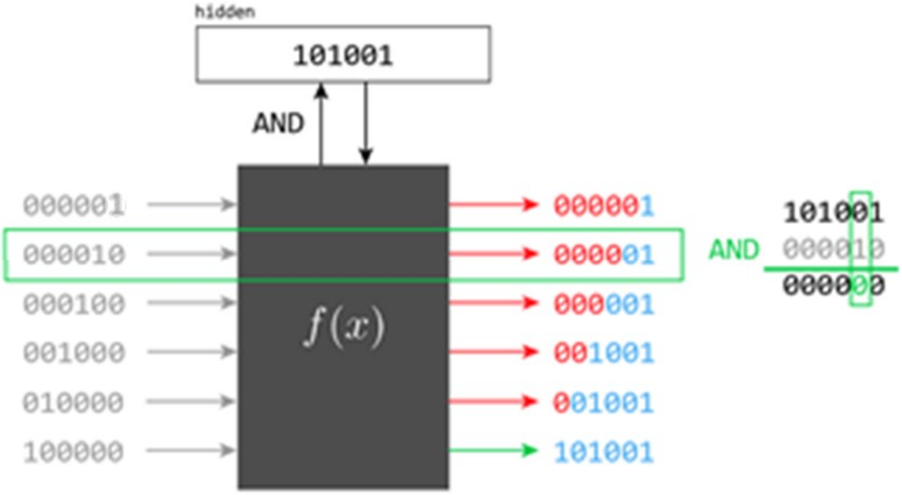


Figure 11. Classical approach to Bernstein Vazirani algorithm.

As opposed to the classical solution, the equivalent quantum algorithm can theoretically output the above just one try. Suppose the above hidden string is reused. The string is 6 bit long, so the inputs are represented by 6 qubits. In general, for n-bit long string, the following superposition is presented. Equation (59) is a result of a transformation after Hadamard gate. Equation (60) shows the transformation when α is 0.

$$|a\rangle \rightarrow \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{ax}|x\rangle\,, \tag{59}$$

$$|00\,\ldots\,0_n\rangle \rightarrow \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle\,. \tag{60}$$

The output is initialised with $|-\rangle$. With a phase kickback trick, the transformation is resulted in Equation (61).

$$|x\rangle \rightarrow (-1)^{s\cdot x}|x\rangle, \tag{61}$$

Results appearing outside the oracle will undergo another set of Hadamard gates (same number as ones applied to input) which will then produce the hidden string s. Figure 12 below shows what is currently in the Oracle. Note that the hidden string currently is 101001, and that we do not know the value nor how the Oracle is structured.

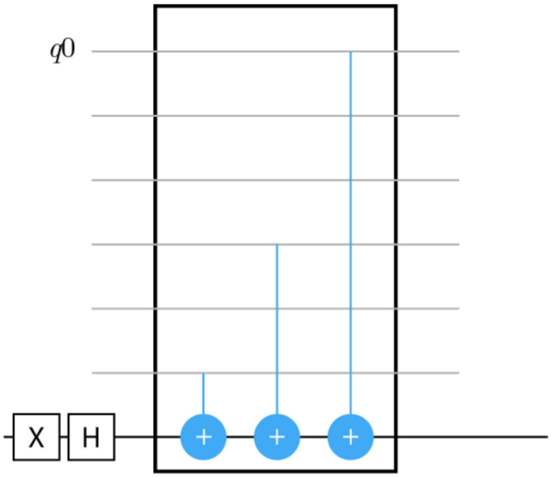


Figure 12. Illustration of the formation within the Oracle.

Since all inputs are 0, upon contact with the blue lines, these inputs will turn into 1. Once measured, the hidden string is effectively found. This is done only once, as no iterations are required. Figure 13 illustrates a general idea of the implementation with three stages: (1) transformation after Hadamard gates, (2) result corresponding to the hidden string, (3) result which had undergone measurement.

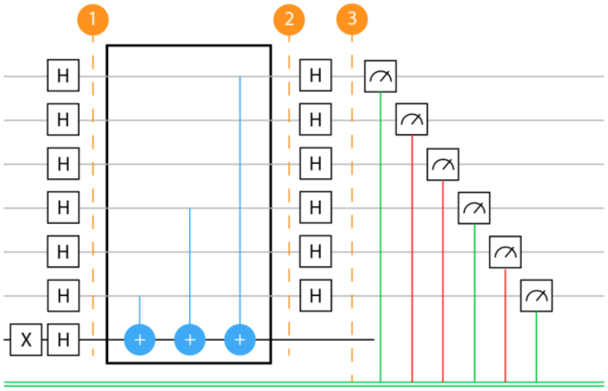
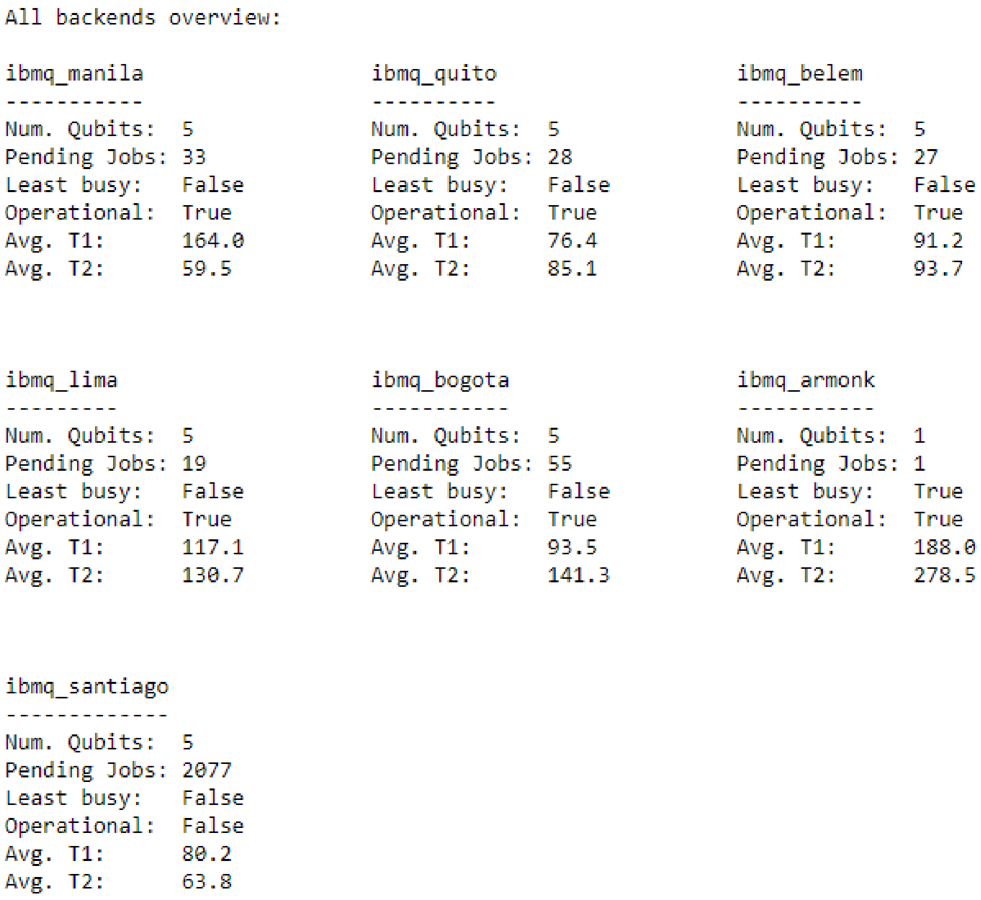


Figure 13. Illustration of the Bernstein-Vazirani implementation.

VI. METHODOLOGY

All quantum algorithms will be assisted by the backends of IBM Quantum Computing. As of this paper, there are 7 backends available for use, as shown in Figure 14. Quantum algorithms are heavily involved in probabilities, and all backends allow a maximum of 8192 runs of a single algorithm, known as shots, for a probability distribution of results. Every implementation is done through Python v3.9.7, along with Qiskit.



```
All backends overview:

ibmq_manila                  ibmq_quito                   ibmq_belem
-----------                  ----------                   ----------
Num. Qubits:  5              Num. Qubits:  5              Num. Qubits:  5
Pending Jobs: 33             Pending Jobs: 28             Pending Jobs: 27
Least busy:   False          Least busy:   False          Least busy:   False
Operational:  True           Operational:  True           Operational:  True
Avg. T1:      164.0          Avg. T1:      76.4           Avg. T1:      91.2
Avg. T2:      59.5           Avg. T2:      85.1           Avg. T2:      93.7



ibmq_lima                    ibmq_bogota                  ibmq_armonk
---------                    -----------                  -----------
Num. Qubits:  5              Num. Qubits:  5              Num. Qubits:  1
Pending Jobs: 19             Pending Jobs: 55             Pending Jobs: 1
Least busy:   False          Least busy:   False          Least busy:   True
Operational:  True           Operational:  True           Operational:  True
Avg. T1:      117.1          Avg. T1:      93.5           Avg. T1:      188.0
Avg. T2:      130.7          Avg. T2:      141.3          Avg. T2:      278.5



ibmq_santiago
-------------
Num. Qubits:  5
Pending Jobs: 2077
Least busy:   False
Operational:  False
Avg. T1:      80.2
Avg. T2:      63.8
```

Figure 14. Available IBMQ backends with number of qubits and other information listed.

In Bernstein-Vazirani algorithm, 4 different size of secret binary string is randomly generated, 4 quantum circuits with various secret string are constructed based on the details mentioned in section V.C. Hence one secret string will be corresponding to one quantum circuit. The quantum circuits are executed using simulator and real quantum device from IBM, thus the time taken for each circuit are documented. For the classical approach to this problem, the execution time is also documented. Three types of execution time are visualized by a time complexity diagram, where size and time are the x and y axis respectively.

In the Grover's algorithm, a set of values will be taken for searching, namely $|10\rangle$,$|100\rangle$,$|1000\rangle$,$|10000\rangle$, for size of 2, 3, 4 and 5 qubits respectively. For the classical approach, the fastest searching algorithm, binary search is used, using the same set of numbers, sorted to emphasise only on binary's search time complexity. For each case, the runtime, measured in seconds will be tabulated and compared. The runtime results will be collected over five experiments with 1024 shots and averaged for a runtime which represents each algorithm. For consistency, the quantum backend imbq_lima will be used for all Grover's algorithms.

VII. DISCUSSION AND RESULTS

Results of the linear search against Grover's algorithm were recorded and tabulated in Table 3. There were two results tabulated for the latter algorithm: (1) QASM simulator to emulate ideal execution of a quantum circuit, and (2) IBMQ backend imbq_lima for real execution. All results were taken in seconds.

Table 3: Results of linear algorithm vs. Grover's algorithm

Sample size (qubits)	**Linear search**	**Grover's algorithm (1)**	**Grover's algorithm (2)**
2	0.0005878040	0.0010023117	2.914091825
3	0.0006287839	0.0019998550	4.173945904
4	0.0007690690	0.0020256042	4.698157549
5	0.0010677010	0.0029685497	5.570025584

Visualization of the above results are shown in Figure 15,

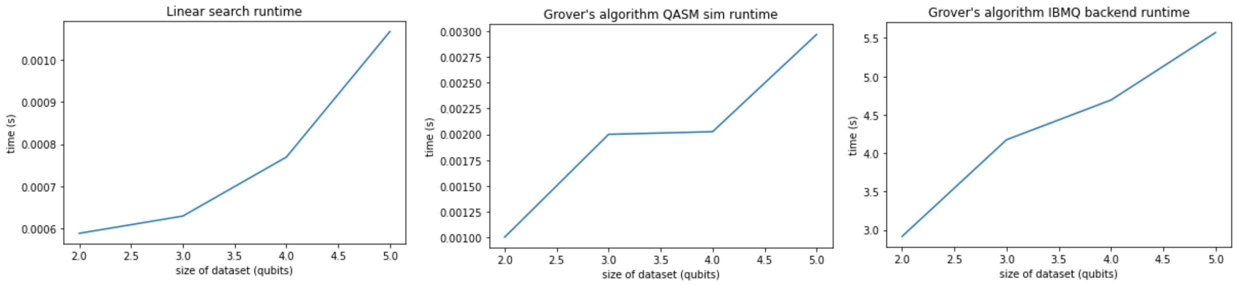


Figure 15. Runtime of Grover's algorithm with different approach

Interestingly, linear search had shown faster results in runtime than both the QASM simulation and IBMQ backend. Scott explained this in terms of relativity [19]. The proven time complexity of Grover's algorithm, which is $\sqrt{n}$ is only applicable for when the oracle is ready to process. However, the algorithm sent to the backend required loading and prior computation of the experiment, which resulted in the extra duration that was taken.

Hardware used for the linear search algorithm was a system equipped with an Intel i5 10400F with 16 GB RAM, whereas for the Grover's algorithm, imbq_lima had a maximum potential of 5 qubits only.

Results of both quantum and classical approaches of Bernstein-Vazirani algorithm are tabulated in Table 4. Two results are produced for quantum approach, where the first result is for simulator, and the second result is for real quantum computer at IBMQ backend ibmq_belem, it is selected for being the least busy backend. All results were taken in seconds. Note that we need an auxiliary qubit for this implementation, and since IBM's backend only provide maximum of 5 qubits per backend. Thus, secret string with 5 bits is not examine in our project.

Table 4: Results of Classical approach vs. Bernstein-Vazirani algorithm

Sample size (qubits)	**Classical approach**	**Bernstein-Vazirani algorithm (1)**	**Bernstein-Vazirani algorithm (2)**
1	1.73e-6	0.0030066967	3.154468774
2	4.41e-6	0.0029921532	3.617537021
3	8.12e-6	0.0029883385	4.122985839
4	13e-6	0.0020451546	4.183508157

Visualization of the above results are shown in Figure 16.

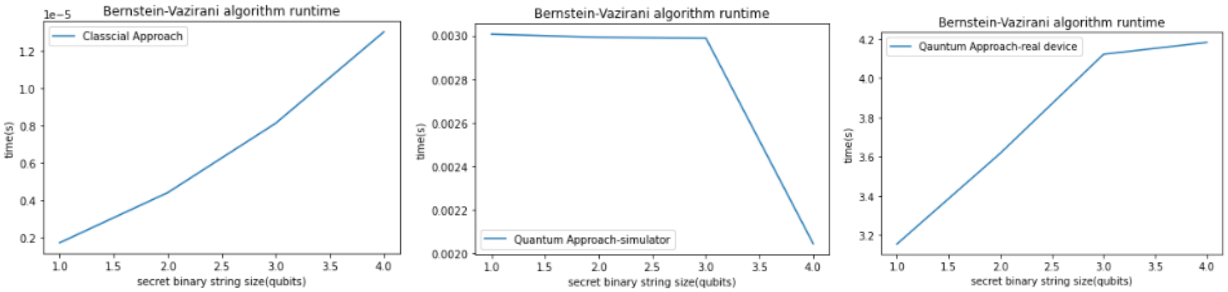


Figure 16. Runtime of Bernstein-Vazirani algorithm with different approach.

By observation of the diagrams, classical approach has a linear trend as expected from the theoretical time complexity $O(n)$. While both results from quantum approach show a nearly constant trend, which is also expected due to the constant time complexity from previously mentioned analysis. Note that if more qubits are allowed for manipulation, the quantum approach will better fit the constant time complexity model.

Although the theoretical time complexity of the quantum approach is constant, the time taken for quantum computing is still at a higher cost compared to classical computers, the results showed that quantum computing in practice still requires large steps of improvements. Hardware for quantum simulation and classical approach is Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz CPU with 8GB RAM.

VIII. LIMITATIONS

Ibmq backends at the time of this study are equipped with a maximum of 5 qubits. A large dataset, for example 20 qubits or more would have been ideal for the analysis of runtime between both algorithms. A quantum circuit with more than 4-qubit was impossible due to the limitations of coupling map. Increasing the number of qubits increases the depth of the circuit in terms of gate manipulations. The error rate for a basis gate of a qubit is small, but the cumulative errors of many of these gates will raise these error rates to a significant level. The accuracy of execution time was also limited by the IBM machines running the algorithms. Running 100 experiments or more for accurate data will be very time consuming due to the nature of time-sharing with other users, where a queue-up is needed.

IX. CONCLUSION

In this paper, the concepts of quantum gates are briefly discussed. The runtime of a quantum algorithm as compared to the classical algorithm was analysed, and results suggest that current hardware is still not suitable for circuits for higher-qubit quantum circuit implementation.

From Table 5 and 6. we can see that the differences in computing time are in the scale of thousands and even hundred thousand.

Table 5: Comparison of classical and Grover's algorithm

Sample size (qubits)	Linear search	Grover's algorithm (2)	Factor of time difference
2	0.0005878040	2.914091825	4957.591
3	0.0006287839	4.173945904	6638.125
4	0.0007690690	4.698157549	6108.89
5	0.0010677010	5.570025584	5216.84

Table 6: Comparison of classical and Bernstein-Vazirani algorithm

Sample size (qubits)	Classical approach	Bernstein-Vazirani algorithm (2)	Factor of time difference
1	1.73e-6	3.154468774	1823392
2	4.41e-6	3.617537021	820303.2
3	8.12e-6	4.122985839	507756.9
4	13e-6	4.183508157	321808.3

Although the results in Figure 15 and Figure 16 showed a linear trend of the time complexity of the quantum algorithms, classical algorithms still run much faster due to well-developed performance enhancements techniques, the results match the conclusions conducted by other papers [7].

On the other hand, a Qiskit package is developed to provide instant access for users that want to perform the same benchmarking introduced in this paper, with some degree of customizability such as rounds of testing, and number of qubits. For the time being, the proposed package contains only two algorithms, however, more quantum algorithms will be added to the package in the future. Detailed instructions and usages can be found in the Github repository listed in the following section.

APPENDIX

Source code can be found in the following repository: https://github.com/Youngei402tw/QuantumComputing

ACKNOWLEDGEMENT

The authors are extremely grateful for Qiskit's platforms and tools. The conclusions discussed in this paper are only comments from the authors, and do not reflect the opinions of the Qiskit team. This paper was not funded.

REFERENCES

[1] P. Abhishek, and V. Ramesh, "Quantum computing for big data analysis", Indian Journal of Science, 2015, 14(43), pp. 98-104.

[2] A. Frank, A. Kunal, B. Ryan, B. Dave, B. C. Joseph, et al. "Quantum supremacy using a programmable superconducting processor", Nature 574, 2019, pp. 505 - 51.

[3] Y. Wu, W.-S. Bao, S. Cao, F. Chen, M.-C. Chen, et al. "Strong quantum computational advantage using a superconducting quantum processor", Physical Review Letters, 2019, 127(18).

[4] V. Chamola, A. Jolfaei, V. Chanana, P. Parashari, and V. Hassija, "Information security in the Post Quantum era for 5G and beyond networks: Threats to existing cryptography, and post-quantum cryptography," Computer Communications, vol. 176, pp. 99–118, 2021.

[5] G. Acampora, and A. Vitiello, "Implementing evolutionary optimization on actual quantum processors." Information Sciences, 2021, 575, pp. 542–562.

[6] H. Soeparno, and A. S. Perbangsa, "Cloud quantum computing concept and development: A systematic literature review." Procedia Computer Science, 2021, 179, pp. 944–954.

[7] L. Gyongyosi, and S. Imre, "A survey on quantum computing technology." Computer Science Review, 2019, 31, 51–71.

[8] G. A. Nemnes and D. Dragoman, "Reconfigurable quantum logic gates using Rashba controlled spin polarized currents," Physica E: Low-dimensional Systems and Nanostructures, vol. 111, pp. 13–19, 2019.

[9] A. Pourkia and J. Batle, "Cyclic groups and quantum logic gates," Annals of Physics, vol. 373, pp. 10–27, 2016.

[10] H. M. Gaur, A. K. Singh, and U. Ghanekar, "In-depth comparative analysis of reversible gates for designing logic circuits," Procedia Computer Science, vol. 125, pp. 810–817, 2018.

[11] R. M. Abdullah, R. Basher, A. A. Alwan, and A. Z. Abualkishik, "Quantum Computers for optimization the performance," Procedia Computer Science, vol. 160, pp. 54–60, 2019.

[12] M. L. Dalla Chiara, R. Giuntini, G. Sergioli, and R. Leporini, "A many-valued approach to quantum computational logics," Fuzzy Sets and Systems, vol. 335, pp. 94–111, 2018.

[13] E. Bäumer, "Qubits and Quantum States, Quantum Circuits, Measurements - Part 1," in *1. Qubits and Quantum States, Quantum Circuits, Measurements - Part 1*, 2021

[14] E. Bäumer, "Qubits and Quantum States, Quantum Circuits, Measurements - Part 2," in *2. Qubits and Quantum States, Quantum Circuits, Measurements - Part 2*, 2021

[15] E. Bäumer, " Qubits and Quantum States, Quantum Circuits, Measurements - Part 3," in *3. Qubits and Quantum States, Quantum Circuits, Measurements - Part 3*, 2021

[16] C. de Ronde, The (Quantum) Measurement Problem in Classical Mechanics , 2020.

[17] R. G. Littlejohn, "The Mathematical Formalism of Quantum Mechanics," 2020

[18] L. Vaidman, "Derivations of the born rule," *Jerusalem Studies in Philosophy and History of Science*, pp. 567–584, 2020. doi:10.1007/978-3-030-34316-3_26

[19] S. Aaronson, "Doing my oracle duty," Shtetl-Optimized, https://scottaaronson.blog/?p=451 (accessed Jul. 15, 2023).

[20] A. J. Daley *et al.*, "Practical quantum advantage in quantum simulation," *Nature*, vol. 607, no. 7920, pp. 667–676, 2022. doi:10.1038/s41586-022-04940-6

[21] M. Schuld and N. Killoran, "Is quantum advantage the right goal for quantum machine learning?," *PRX Quantum*, vol. 3, no. 3, 2022. doi:10.1103/prxquantum.3.030101

[22] M. Cerezo, G. Verdon, H.-Y. Huang, L. Cincio, and P. J. Coles, "Challenges and opportunities in quantum machine learning," *Nature Computational Science*, vol. 2, no. 9, pp. 567–576, 2022. doi:10.1038/s43588-022-00311-3

[23] C. W. Bauer *et al.*, "Quantum simulation for high-energy physics," *PRX Quantum*, vol. 4, no. 2, 2023. doi:10.1103/prxquantum.4.027001