Shor's Algorithm simulation

by

Nont Arayarungsarit

A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Engineering in
Computer Science

Examination Committee: Prof. Chantri Polprasert

Mrs. Sandhya Lamichhane

Nationality: Thai

Previous Degree: Bachelor of Engineering in Electrical Engineering (Thai program), Kasetsart University, 50 Ngamwongwan Rd, Khwaeng Lat Yao, Khet Chatuchak, Bangkok, Thailand

Scholarship Donor: AIT Scholarship

Asian Institute of Technology
School of Engineering and Technology
Thailand
Feb 2024

ACKNOWLEDGEMENT

I am grateful to professor Chantri Polprasert for arousing my interest in the Algorithm Design and Analysis course (ADA) to encourage many papers with source code for studying mechanism of algorithm.

CONTENT

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

In historical problem-solving endeavors, humanity initially tackled issues through manual processes, utilizing reasoning and cause-and-effect methodologies. As challenges escalated from basic to intricate, mathematics emerged as a powerful tool for problem resolution. However, as the complexity of problems continued to grow, relying solely on manual computation became increasingly impractical. The integration of computers, particularly quantum computers, became a transformative solution to address these challenges.

Contemporary computing systems employ a binary representation (0 and 1) to store information in the form of bits. Despite this advancement, solving intricate problems still encounters limitations, particularly in terms of time complexity. The emergence of quantum computers, a milestone marked in the 1990s, introduced a paradigm shift by utilizing qubits or quantum bits to represent information. Unlike classical bits, qubits can exist in a superposition of both 0 and 1 states simultaneously, allowing for parallel computation and significantly enhancing computational efficiency. At the core of efficient problem-solving lie algorithms, and one of the groundbreaking algorithms is Shor's algorithm, conceived by Peter Shor in 1994. Shor's algorithm has garnered attention for its proficiency in factoring large numbers, a computationally challenging task for classical computers. Quantum computers, when equipped with Shor's algorithm, demonstrate an unprecedented capability to factorize large numbers within minutes, a task that would take classical computers millions of years to accomplish.

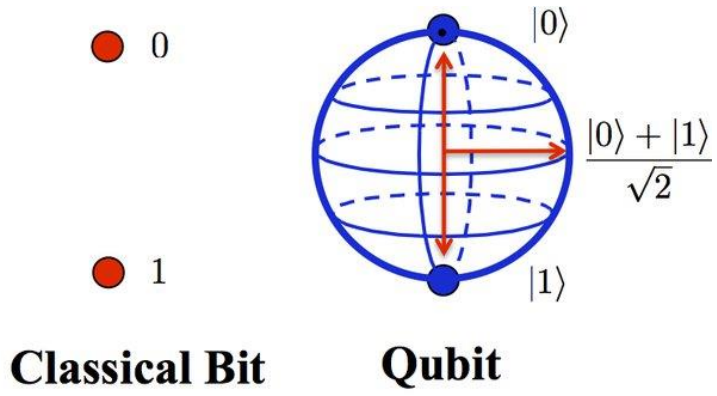


Figure 1.1 : The different between classical bit (Classical Computer) and qubit (Quantum Computer)

Ref : https://www.quora.com/What-is-the-difference-between-a-quantum-and-a-classical-computer

This thesis endeavors to delve into the intricacies of Shor's algorithm, elucidating the underlying processes, steps, and algorithms involved in simulating the factorization of large numbers. However, it's essential to acknowledge the inherent limitations of classical computers, prompting a focus on simulating the factorization of moderately large numbers rather than venturing into the realm of extremely large ones. Through this exploration, we aim to shed light on the revolutionary potential that quantum computing holds for addressing complex problem-solving scenarios in a time-efficient manner.

1.2 Statement of the Problem

Quantum computing, marked by the advent of Shor's algorithm, represents a transformative leap in addressing computationally complex mathematical challenges. One of the algorithm's primary problem statements is rooted in the classical computer's incapacity to efficiently factorize large numbers. Traditional methods for factorization on classical computers exhibit an impractical time complexity, rendering them inadequate for handling integers of significant magnitude. Shor's algorithm, however, leverages the principles of quantum mechanics, introducing the concept of qubits that can exist in multiple states simultaneously. This quantum parallelism enables the algorithm to achieve factorization at an unprecedented speed, revolutionizing the landscape of mathematical problem-solving.

At the heart of Shor's algorithm lies the ingenious approach to factorizing large numbers by exploiting quantum superposition and entanglement. The problem addressed is not merely the factorization itself but the inefficiency of classical computers in executing this task within reasonable time frames. Shor's algorithm demonstrates that with quantum computing, factorization becomes a significantly more tractable problem. This problem statement underscores the algorithm's potential to revolutionize cryptographic systems, as many encryption techniques rely on the difficulty of factorizing large numbers for their security.

Furthermore, the significance of Shor's algorithm extends beyond its immediate problem of efficient factorization. It poses a broader challenge to classical computing paradigms, signaling a paradigm shift in the way we approach mathematical problem-solving. The algorithm prompts us to reconsider the limitations imposed by classical computational methods and explore the vast potential of quantum computing in addressing problems deemed insurmountable within classical frameworks. In essence, Shor's algorithm not only addresses a specific computational challenge but also beckons a reevaluation of our entire approach to complex problem-solving in the realm of mathematics and cryptography.

1.3 Objectives

1.3.1 Be able to simulate, understand and implement mechanism of Shor's algorithm.

Organization of the Study

1.4.1 IBM has been actively working on quantum computing research and development. IBM's quantum computing efforts are centered around their IBM Quantum program, which aims to advance quantum computing technologies and make them accessible to the broader scientific and business communities.

1.4.2 Google has been actively involved in quantum computing research. Google's quantum computing efforts are primarily focused on developing and advancing quantum processors, algorithms, and applications.

1.4.3 IonQ is a company that focuses on developing quantum computers based on trapped-ion technology. Trapped ions are individual ions that are trapped and manipulated using electromagnetic fields to perform quantum operations. IonQ has been working on advancing the field of quantum computing and making quantum processors available to users.

1.4.4 QCI is a company that focuses on providing quantum computing solutions and services. QCI is involved in developing quantum algorithms, software, and tools to leverage the capabilities of quantum computers for solving complex problems.

CHAPTER 2

LITERATURE REVIEW

2.1 Quantum Computer

A real quantum computer is physically composed of three major components. A conventional computer and its supporting hardware do programming and communicate commands to the qubits in the first section. A technique for sending signals from the computer to the qubits is covered in the second section. The qubits must, at last, be stored somewhere. Certain demands or conditions must be satisfied, and this qubit storage unit must be able to stable the qubits and there is an experiment that can prove the quantum mechanics which can exist in quantum computer.

When we shine the light by shooting photon pellets through double-slits, we then check using some of sensors to see which photon can pass through to which slits. How much or how little is there? And where did the photons hit the scene? After the experiment, it was found that there can classified to 2 groups of photons that could pass through the slits. This experiment can implement superposition property as well.

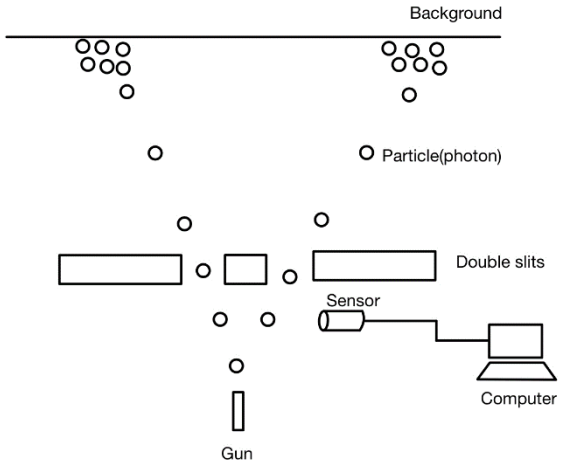


Figure 2.1 : When measuring the photon, it can be classified into two groups (superposition property)

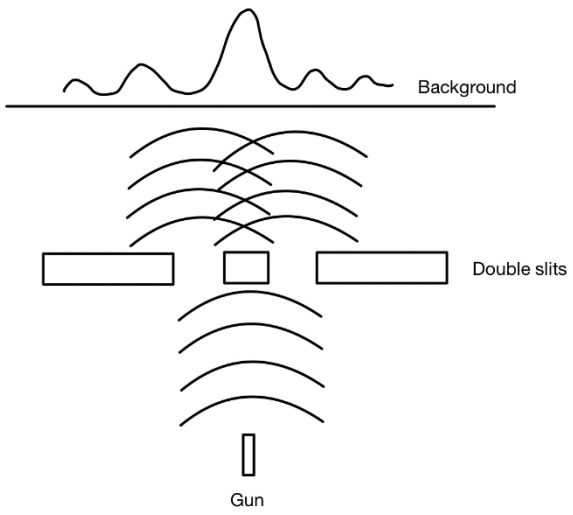


Figure 2.2 : No sensor and computer for measurement, it can act like interference of waves (superposition property)

which after inspection and measurement found that the assumptions are different from reality. Completely, making it possible to conclude that When we do not measure or inspect with photon it will be like the interference of waves but if those photons are measured or examined will cause the photon grains to affect on the background. Photon will try all the possible paths as they can. Each photon has superposition property and entanglement property as follows

2.1.1 Superposition state will collapses under observation to reveal single point on the screen and if the paths have already exist, it will be appear to the background of slit. They can take multiple paths simultaneously and exhibit wave-like interference patterns. This concept is a fundamental aspect of quantum mechanics, where particles can exist in multiple states until measured.

2.1.2 Entanglement is another key quantum phenomenon. Entanglement occurs when two or more particles become correlated in such a way that the state of one particle instantaneously influences the state of the other(s), regardless of the distance between them. The concept of entanglement is often discussed in the context of quantum entanglement experiments. This can be describe in another way by if the unknown value forced and specific to be one value, it will related to another value as well.

Although, all of two properties above can apply to quantum computer's working by solving some problems that have all the exist paths or values (not Bruce force but superposition and entanglement), then we can specific one path or one value and the rest of the path or value will automatically collapse. Finally, the quantum computer can extremely solve the problem faster than classical computer. Moreover, Quantum computers delve into the frigid realm of temperatures just above absolute zero, around -273.15 Celsius, fostering an environment where certain materials showcase the intriguing phenomenon of superconductivity. In this state, electrical resistance drops to zero (R=0), not only enabling efficient current flow but also creating a conducive setting for the manipulation of quantum bits, or qubits. These qubits, governed by the principles of superposition and entanglement, exhibit the unique ability to seamlessly transition between states with remarkable speed and efficiency.

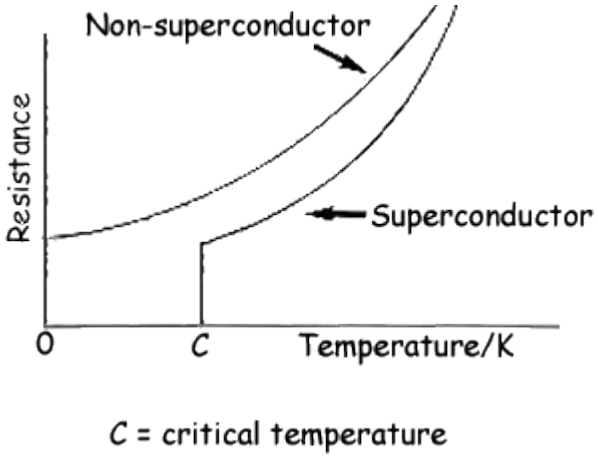


Figure 2.3 : Relation between resistance and temperature of Superconductors and Superconducting Materials Information

ref :
https://www.globalspec.com/learnmore/materials_chemicals_adhesives/electrical_optical_specialty_materials/superconductors_superconducting_materials

The literature also delves into the theoretical underpinnings of quantum mechanics that enable quantum computers to outperform classical computers in specific tasks. Principles such as superposition and entanglement, which allow qubits to exist in multiple states simultaneously, play a pivotal role in the functioning of algorithms like Shor's. Researchers have extensively explored the mathematical foundations and quantum gates that form the basis of quantum computation, providing a comprehensive understanding of the computational advantages offered by quantum systems.

2.2 Shor's algorithm

Quantum computing, with its groundbreaking algorithms like Shor's algorithm, has become a focal point in the realm of computational science and cryptography. The literature on quantum computing extensively explores the fundamental principles, algorithms, and applications that distinguish quantum computation from classical methods. "Quantum Algorithm Implementations for Beginners" introduced the algorithm that revolutionized the field by efficiently factoring large numbers. Shor's algorithm stands out as a testament to the quantum advantage, showcasing its ability to solve complex mathematical problems exponentially faster than classical computer or binary computer.

In the context of cryptography, the literature review highlights the implications of Shor's algorithm for widely used encryption methods like RSA encryption. Classical encryption techniques, relying on the difficulty of factorizing large numbers, face a significant threat from quantum computers equipped with Shor's algorithm. This has prompted discussions on the need for post-quantum cryptography, exploring alternative encryption methods resistant to quantum attacks. The literature review thus reflects a dynamic landscape, where the intersection of quantum computing and cryptography necessitates ongoing research and development to address emerging challenges and harness the potential of quantum technologies responsibly. The more qubits that quantum computer have, the more accuracies and quickness to solve the problems.

2.2.1 Structure of Shor's algorithm

Shor's algorithm consist of 2 parts, classical part which can be done in binary computer and quantum part. For the quantum part will find suitable number of period

$$f(x) = g^x \bmod N$$

g : guess number from classical part. This must be coprime number with N

x : the number which begin with 0,1,2,3,…

N : Input number that want to be factorized

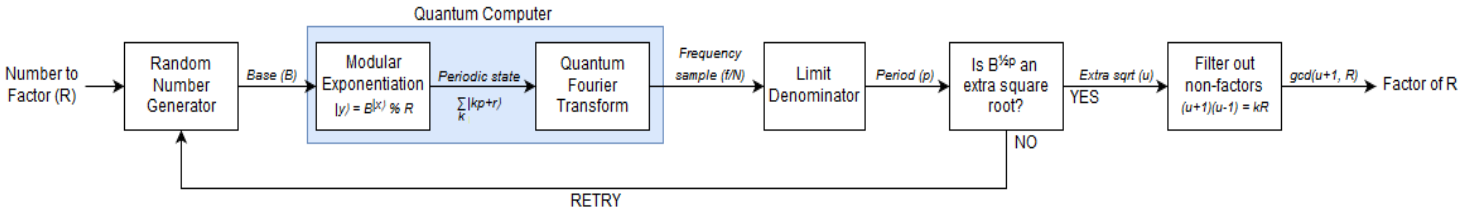


Figure 2.4 : Structure of Shor's algorithm

More over, to find the period of function with Quantum Fourier Transform (QFT), The algorithm will have more efficiency enough to find the period. Basically, The quantum Fourier transform is a linear transformation on quantum bits, and is the quantum analogue of the discrete Fourier transform.

Like the classical Fourier transform, quantum Fourier transform takes data from the original signal representation to the frequency domain representation. The QFT differs from the classical Fourier transform in that it operates on a superposition state and produces a different superposition state as the output.

The quantum Fourier transform can be performed efficiently on a quantum computer, with a particular decomposition into a product of simpler unitary matrices. Using a simple decomposition, the discrete Fourier transform on $O(2^n)$ amplitudes can be implemented as a quantum circuit consisting of only $O(n^2)$ Hadamard gates and controlled phase shift gates, where n is the number of qubits. This can be compared with the classical discrete Fourier transform, which takes $O(n2^n)$ gates which is exponentially more than $O(n^2)$. However, the quantum Fourier transform acts on a quantum state, whereas the classical Fourier transform acts on a vector, thus not every task that uses the classical Fourier transform can take advantage of this exponential speedup.

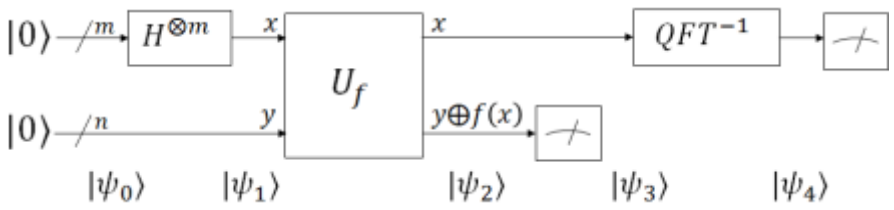
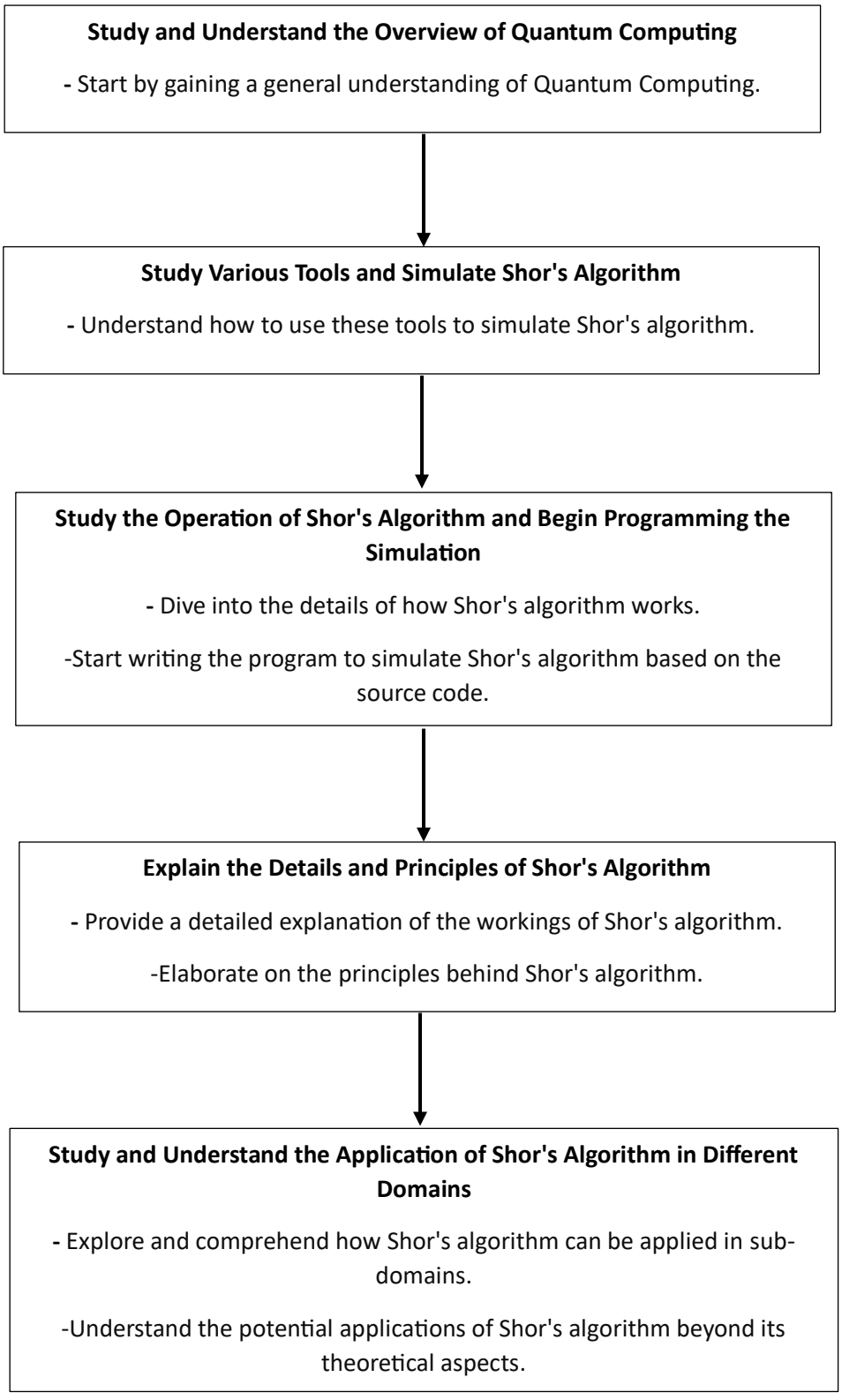


Figure 2.5 : Quantum circuit to find the period of the function f(x) = $g^x \ mod \ N$

CHAPTER 3

METHODOLOGY

3.1 Proposed Methodology



Pseudo code :

```
Receive input number(N) from user

g = Random number from 2 to N

GCD = find gcd (greatest common divider) between g and N

        if(GCD != 1)
                print(N can be factorized : ? x ?)
        if(GCD == 1)
                for(x=0;x<=N;x++) //In this section Quantum can find period r effectively
                        y = pow(a,x) % N
                        append y to the linked list
                        if(first element of array == x^th element of linked list)
                                r = x
                                if(r%2!=0)
                                        random new g
                                if(pow(a,r)%N == -1)
                                        random new g
                if(pow(a,r/2)%N==-1)
                        random new g

p = gcd((pow(g,r/2)+1),N)
q = gcd((pow(g,r/2)-1),N)
print(N can be factorized : p * q)
```

Reason to choose Shor's algorithm

Shor's algorithm can implement and develop some logics of programing combine with physics and electrical knowledges background especially, quantum mechanics, interference of particle or wave length, logic gates circuit, Fourier transform etc. thus Shor's algorithm can gather and apply all of this knowledges with many field of studies.

Benefits and Drawbacks of Shor's algorithm

Benefits	Drawbacks
Optimization and Simulation Shor's algorithm can be applied to solving complex optimization problems, such as the traveling salesman problem or network routing optimization. It can also enable more efficient simulations of quantum systems, leading to advancements in material science, drug discovery, and optimization of physical processes.	**Error Correction** Quantum computations are highly susceptible to noise and errors. Developing efficient error correction techniques and fault-tolerant quantum systems is critical for accurate and reliable execution of Shor's algorithm.
Quantum Error Correction Shor's algorithm provides crucial insights into quantum error correction codes, which are vital for preserving the integrity of quantum computations in the presence of noise and errors. Effective error correction is essential for building large-scale, fault-tolerant quantum computers.	**Hardware Limitations** Building and maintaining stable qubits required for executing Shor's algorithm remains a formidable task. Overcoming hardware limitations and scaling up quantum computers is essential for realizing the full potential of the algorithm.
Number Theory and Mathematics Shor's algorithm has deep connections to number theory and mathematical concepts, driving advancements in these fields. It has contributed to a deeper understanding of prime numbers, modular arithmetic, and the structure of algorithms.	**Security Risk** If Quantum Computer can break RSA encryption by Shor's algorithm, some of confidential data will be faster decrypted and leaked of confidential data.

Steps can be described as follows

1.) Choose a random number g is picked, such that 2 < g < N. Compute gcd(g, N).

2.)This can be done using the Euclid Algorithm.

3.)If gcd(g, N) != 1, then there is a non-trivial factor of N.

If gcd(g,N) == 1, then seek y = g^x + p mod N and find the period of y.

4.)If r is odd, then go back to Step 1.

5.)If $g^{\frac{r}{2}}$ mod N == -1, then go back to Step 1.

6.)The gcd($g^{\frac{r}{2}} + 1, N$) gcd($g^{\frac{r}{2}} - 1, N$) are a non-trivial factor of N

Sample of Dataset to test algorithm

Basically, the number to factor will approximately around 100. This can describe the dataset as following table

Input number	Output number
15	5 x 3
21	7 x 3
35	7 x 5
77	11 x 7

Criteria of number to be tested

1. The number to be factored (n) must be >= 15

2. The number to be factored must be odd.

3. The number must not be prime.

4. The number must not be a prime power.

Source code :

https://github.com/Nont18/Shor-s-algorithm-simulation

CHAPTER 4

RESULTS

Preliminary results time complexities

By analyzing the algorithm that have developed, we can explain that the number as a input have more value, it will give more period as well. Based on the knowledge as we known, when period is huge, the pointer will try to find and check the same value from header node that match with it or not so we can use number of input plotted as x-axis and period as y-axis to represent the tendency of time complexities of Shor's algorithm.

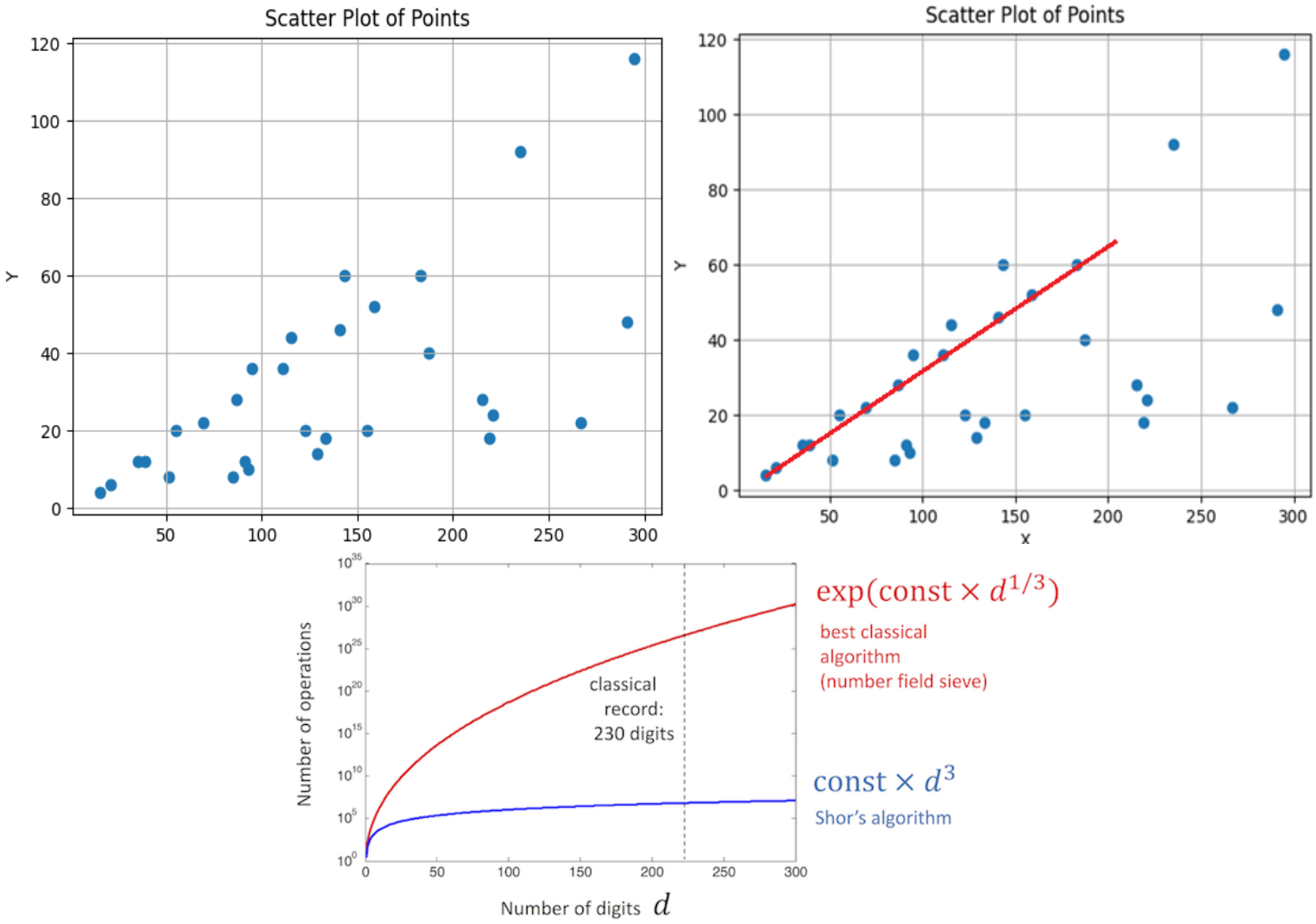


Figure 4.1 : The relation between number (x-axis) and period (y-axis) (above) and relation between number of digits and number of operations (below)

From above figure, If we connect the dot by drawing the line that have most point passed, it show that it would be a linear time complexities. Moreover, the algorithm may depends on the data structure that we have use to implement the algorithm. In this case, we have use linked list, a linear data structure, to store the number of period instead of other data structure and we did not simulate the real qubit but just only implemented the steps so the comparison between time complexities of algorithm that we have developed and time complexities that other claim, the tendency between two graphs has a different one but the same conclusion is with more number of input, the number to be factored will get more time to solved.

CHAPTER 5

CONCLUSION

Although the algorithm that we have developed, it can well implement some part of Shor's algorithm but there have a few error occur during factorizing for example, some number still cannot be factored or it can factored but the answer is not correct like 65. The main problem can tell us that the algorithm is not well developed enough to reach all the possible number but I already check that most of number is correct.

```
Welcome to the simulation of Shor's algorithm.
There are four restrictions for Shor's algorithm:
1) The number to be factored (n) must be >= 15.
2) The number to be factored must be odd.
3) The number must not be prime.
4) The number must not be a prime power.
Enter N to factorize: 65
N can be factorized by 65 X 1
g=2
(pow(g, r / 2) + 1) = 0
(pow(g, r / 2) - 1) = 0
r = 12
List: 1, 2, 4, 8, 16, 32, 64, 63, 61, 57, 49, 33, 1, 2, 4, 8, 16, 32, 64, 63, 61, 57, 49, 33, 1, 2, 4, 8, 16, 32, 64, 63, 61, 57, 49, 33, 1,
 2, 4, 8, 16, 32, 64, 63, 61, 57, 49, 33, 1, 2, 4, 8, 16, 32, 64, 63, 61, 57, 49, 33, 1, 2, 4, 8, 16,
PS C:\Users\araya\Desktop\vscode\shor_dev\real algorithm>
```

Figure 5.1 : The defect of algorithm that I already developed

```
There are four restrictions for Shor's algorithm:
1) The number to be factored (n) must be >= 15.
2) The number to be factored must be odd.
3) The number must not be prime.
4) The number must not be a prime power.
Enter N to factorize: 77
N can be factorized by 11 X 7
g=2
(pow(g, r / 2) + 1) = 0
(pow(g, r / 2) - 1) = 0
r = 30
List: 1, 2, 4, 8, 16, 32, 64, 51, 25, 50, 23, 46, 15, 30, 60, 43, 9, 18, 36, 72, 67, 57, 37, 74, 71, 65, 53, 29, 58, 39, 1, 2, 4, 8, 16, 32,
 64, 51, 25, 50, 23, 46, 15, 30, 60, 43, 9, 18, 36, 72, 67, 57, 37, 74, 71, 65, 53, 29, 58, 39, 1, 2, 4, 8, 16, 32, 64, 51, 25, 50, 23, 46,
15, 30, 60, 43, 9,
```

```
 -o shor } ; if ($?) { .\shor }
Welcome to the simulation of Shor's algorithm.
There are four restrictions for Shor's algorithm:
1) The number to be factored (n) must be >= 15.
2) The number to be factored must be odd.
3) The number must not be prime.
4) The number must not be a prime power.
Enter N to factorize: 341
N can be factorized by 11 X 31
g=2
r = 10
List: 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 3
2, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1
, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64,
128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4,
 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 25
6, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16,
 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171,
 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64
, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2, 4, 8, 16, 32, 64, 128, 256, 171, 1, 2,
```

Figure 5.2 : The number that can correct factorized

REFERENCES

[1] Author, nidhipr123. (2024 Jan 24). geeksforgeeks. https://www.geeksforgeeks.org/shors-factorization-algorithm/

[2] Quantum Algorithm Implementations for Beginners: Author, ADETOKUNBO ADEDOYIN and team. (2022). Shor's Algorithm for Integer Factorization. Page 29.

[3] V. Bhatia and K. R. Ramkumar, "An Efficient Quantum Computing technique for cracking RSA using Shor's Algorithm," 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India, 2020, pp. 89-94, doi: 10.1109/ICCCA49541.2020.9250806. keywords: {Automation;Conferences;Qubit;Parallel processing;Encryption;Computational efficiency;Quantum Computing;Classical Computing;Entanglement;Superposition;Qubits;Computation time;Shor's Algorithm}

[4] SCIENCEALERT STAFF, "How Do Quantum Computers Work?", https://www.sciencealert.com/quantum-computers

[5] Stephen Gossett, "What Is Quantum Computing?" (Aug 17, 2022), https://builtin.com/hardware/quantum-computing

[6] Brindha Jeyaraman, "The Power of Shor's Algorithm" (June 2, 2023), https://www.linkedin.com/pulse/power-shors-algorithm-brindha-jeyaraman/