Quantum Minimum Searching Algorithm and Circuit Implementation

Yujin Kang
School of Electrical Engineering
Korea University
Seoul, Korea
yujin20@korea.ac.kr

Jun Heo
School of Electrical Engineering
Korea University
Seoul, Korea
junheo@korea.ac.kr

***Abstract*—This paper addresses the quantum minimum searching algorithm and provides simulation results using qiskit library by IBM Q. We propose a whole procedure to implement the quantum minimum searching algorithm with high accuracy and design a quantum circuit for 5-qubit searching. The circuit consists of several oracles(phase inverter) and Grover operators(amplifier) and a comparator. To implement 5-qubit searching, we suggest three things. Modifying the expected number of iterations described in Section II.C, applying the concept of QRAM as an interface between quantum data and classical data in Section II.D, and designing quantum comparator using constant adder by Thomas in 2017 in Section III.A.**

***Index Terms*—Minimum searching algorithm, Quantum searching algorithm, Quantum minimum searching algorithm, Quantum Adder, Quantum comparator**

I. Introduction

Starting from Grover's searching algorithm in [5], we could take a step closer to fully accessing data in quantum space. Thus, we could find a value under special conditions using algorithms extended from Grover's searching algorithm. In this paper, we introduce one of these extended algorithms, *quantum minimum searching algorithm*. Let $T[0 \dots N-1]$ be a large unsorted table of N items and each value extracting from an ordered set defined by the relation $\leq$. The minimum searching problem means that finding a value y such that

$$T[y] \leq T[i], \tag{1}$$

where $0 \leq y \leq N-1$ and $i = 0, \dots, N-1$. In general, $O(N)$ iterations are required to find minimum using the classical algorithm. Using the quantum searching algorithm in [3], we need $O(\sqrt{N})$ iterations to solve this problem. The second section of this paper, *Algorithm*, will provide an overview of Grover's searching algorithm and its extensions, especially the quantum minimum searching algorithm. We suggest modifying the expected total number of iterations because we introduce the concept of QRAM, which registers items to entangled quantum states. We can apply a quantum comparator to these states and observe the register after all Grover iterations. The third section, *Circuit*, will specify the quantum comparator using a quantum constant adder in [6] and provide simulation results of 5-qubit comparator. The fourth section, *Implementation*, will represent the whole procedure and its simulation result. In the final section, *Conclusion*, we will analyze the result of this research, and further suggestions will be covered.

II. Algorithm

A. Grover's searching algorithm

Grover's algorithm searches for a specific subset of items in an unordered set. The algorithm can find the subset quadratically faster than the theoretical limit for classical counterparts. Let $2^n = N$ represent N items in n qubits. In general, the classical algorithm requires N queries in the worst case and Grover's algorithm requires $\sqrt{N}$ queries because an oracle function in Grover's algorithm computes a function simultaneously. This property is called '*Quantum Parallelism*', which makes queries in parallel (see [1]).

Let say we want to find a value x in N items. Starting with a quantum register of n qubits, the state of a quantum register is initialized to an equal superposition of states by applying the Hadamard gate which takes $O(\log_2 N)$ operations. All possible states are mapping to each item.

$$|\psi\rangle = H^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle . \tag{2}$$

The function to find an answer x can be expressed as

$$f(i) = \begin{cases} 1 & \text{if } i \neq x \\ 0 & \text{if } i = x \end{cases}, \tag{3}$$

The oracle function, U_f, and Grover operator, U_G, are shown as (see [5])

$$U_f |i\rangle |q\rangle = |i\rangle |q \oplus f(i)\rangle = (-1)^{f(i)} |i\rangle |q\rangle , \tag{4}$$

where $|q\rangle = |-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$ and

$$U_G |\psi\rangle = (2|\psi\rangle \langle\psi| - I) |\psi\rangle . \tag{5}$$

A Grover iteration consists of an oracle function and a Grover operator. Thus, we can find the value x in N items using Grover iteration.

In the meantime, there is another approach to equations above. According to [2] and [4], finding the value x can be substituted with finding an index i_0 where $T[i_0] = x$ and $0 \leq i_0 \leq N-1$. After j Grover iterations, the state

978-1-7281-6758-9/20/$31.00 ©2020 IEEE

Algorithm 1: Grover's searching algorithm.

```
Initialize $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle$ .
repeat
    Invert a phase using an oracle function, $U_f$.
    Amplify the state below the average amplitude using Grover operator, $U_G$.
until $O(\sqrt{N})$ times: Grover iteration
Measure the state.
return $x$.
```

of a quantum register is expressed in an explicit closed-form formula for real numbers k_j and l_j by [4].

$$|\psi(k_j, l_j)\rangle = k_j |i_0\rangle + \sum_{i=0, i\neq i_0}^{N-1} l_j |i\rangle, \tag{6}$$

$$k_j = \sin((2j+1)\theta), \tag{7}$$

$$l_j = \frac{1}{\sqrt{N-1}} \cos((2j+1)\theta), \tag{8}$$

where $k^2 + (N-1) l^2 = 1$, $\sin^2(\theta) = 1/N$, and j is an integer.

Using (7), we can estimate how many iterations we need to measure i_0 with a high probability of success. To make the probability of $|i_0\rangle$ high, such as 1, the equation (7) satisfies the condition $k_{j_0} = 1$ with j_0 iterations . From the approximation $\theta \approx \sin\theta = 1/\sqrt{N}$ for large N, we can get an equation for the number of iterations j_0.

$$j_0 \approx \frac{\pi}{4}\sqrt{N} = O(\sqrt{N}). \tag{9}$$

Algorithm 1 shows the pseudo-code of Grover's searching algorithm. To simulate Grover's searching algorithm, we need circuit-level implementation. Fig. 1 shows the circuit of Grover's algorithm in [7].

B. Multiple solutions

Grover's searching algorithm can be extended to multiple solutions case. Let t be the number of solutions and $A = \{i|T[i] = x\}$, $B = \{j|T[j] \neq x\}$. Because we amplify the amplitude of correct states which is less than the average amplitude, the number of correct states, such as t in this section, should be less than $N/2$ to make the average amplitude greater than zero.

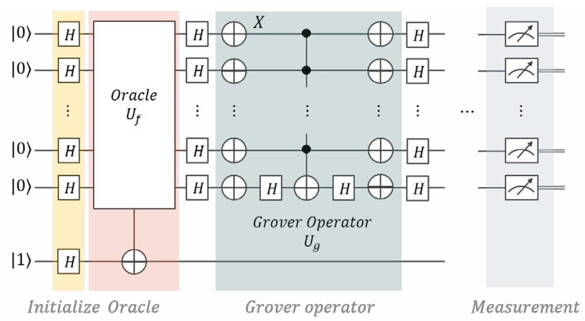


Fig. 1: Circuit implementation of Grover's algorithm.

After j Grover iterations, just as (6), (7) and (8), the state of a quantum register is expressed in an explicit closed-form formula for real numbers k_j and l_j by [4].

$$|\psi(k_j, l_j)\rangle = \sum_{i\in A} k_j |i\rangle + \sum_{i\in B} l_j |i\rangle, \tag{10}$$

$$k_j = \frac{1}{\sqrt{t}} \sin((2j+1)\theta), \tag{11}$$

$$l_j = \frac{1}{\sqrt{N-t}} \cos((2j+1)\theta), \tag{12}$$

where $tk^2 + (N-t) l^2 = 1, \sin^2(\theta) = t/N$ and j is integer.

Using (12), we can estimate how many iterations we need to obtain one of the elements i from A with a high probability of success. By making the probability of the elements from B low, such as 0, we can obtain one of the solutions. From (12), $j_0 = (\pi - 2\theta)/4\theta$ satisfies the condition $l_{j_0} = 0$ and we can use $\theta \approx \sin\theta = \sqrt{t/N}$ for $t \ll N$. Thus, an equation for the number of iterations j_0 to find the solution can be expressed as

$$j_0 \approx \frac{\pi}{4}\sqrt{\frac{N}{t}} = O\left(\sqrt{\frac{N}{t}}\right). \tag{13}$$

The algorithm can be improved by stopping in the middle of iterations and observing the register. If the answer from the register isn't correct, it starts all over again. The expected number of iterations is calculated differently in [4]. However, the expected number of iterations is valid only when $1 \leq t < N/2$. Cases $t > N/2$ and $t = 0$ can be discomposed of by appropriate timeouts.

C. Quantum Minimum Searching algorithm

Using the Grover's searching algorithm in Section II.A and the extended case in Section II.B, the quantum minimum searching algorithm can be defined. Assume we have an unsorted table $T[0 \ \ldots \ \ N-1]$ of N items as same as Section II.B, and we want to find the index y of minimum value $T[y]$ in this table. The quantum minimum searching algorithm takes at most $O_{qmsa}(\sqrt{N})$ iterations with the probability of success at least $1/2$. The algorithm is described in Algorithm 2 and Fig. 2. All labels(1., 2a., 2b., . . .) in Fig.2. are based on Section II in [2].

We can estimate the upper bound of the expected total number of iterations, $O_{qmsa}(\sqrt{N})$, to find the minimum. Assume the algorithm runs until it finds the minimum and that there is no time limit. It is called '*The Infinite Algorithm*'. If we search for items less than the threshold $T[y]$ and the number of those items is k, the probability that we select an item of rank r among k items is given by

$$p(k,r) = \begin{cases} 1/r & \text{if } r \leq k \\ 0 & \text{otherwise} \end{cases} , \tag{14}$$

where r is the rank of the item.

Algorithm 2: Quantum minimum searching algorithm.

```
Choose threshold index $y$ uniformly at random
($0 \leq y \leq N-1$).
$\tau_{total} \leftarrow 0$
repeat
    Initialize $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle$ .
    $\tau_{total} \leftarrow \tau_{total} + \log_2 N$.
    if $T[i] < T[y]$ then
        Oracle marks the items $i$.
    end if
    Apply the quantum searching algorithm(qsa).
    $\tau_{total} \leftarrow \tau_{total} + O_{qsa}(\sqrt{N/t})$.
    Measure the state.
    Observe the register and let $y'$ be the outcome.
    if $T[y'] < T[y]$ then
        $y \leftarrow y'$.
    else
        $y \leftarrow y$.
    end if
until $\tau_{total} \geq O_{qmsa}(\sqrt{N})$.
return  $y$.
```

According to the paper [4], the expected total number of iterations by infinite algorithm is a sum of two terms, the expected iteration in stage 2a and the expected iteration in stage 2b in Fig. 2.

The upper bound of the first one is expressed as

$$\begin{aligned}\sum_{r=2}^{N} p\left(N,r\right)\log_2 N &= \sum_{r=2}^{N} \frac{1}{r}\log_2 N \\ &= \left(H_N - 1\right)\log_2 N \\ &\leq \ln N \log_2 N \\ &= \ln 2 \times \log_2 N \times \log_2 N \\ &\leq \frac{7}{10}\left(\log_2 N\right)^2 ,\end{aligned} \tag{15}$$

where H_N is a Harmonic number which satisfies $H_N = \sum_{k=1}^{N} 1/k$ and $\ln N + 1/N \leq H_N \leq \ln N + 1$. In this paper, the upper bound of the second one is calculated using (13) instead of $9/2\sqrt{N/t}$ in [3] and [4].

$$\begin{aligned}\sum_{r=2}^{N} p(N,r)\frac{\pi}{4}\sqrt{\frac{N}{r-1}} &= \frac{\pi}{4}\sqrt{N}\sum_{r=1}^{N-1}\frac{1}{r+1}\frac{1}{\sqrt{r}} \\ &\leq \frac{\pi}{4}\sqrt{N}\left(\frac{1}{2} + \sum_{r=2}^{N-1}\frac{1}{r\sqrt{r}}\right) \\ &\leq \frac{\pi}{4}\sqrt{N}\left(\frac{1}{2} + \int_{r=1}^{N-1} r^{-\frac{3}{2}}dr\right) \\ &= \left(\frac{5\pi}{8}\sqrt{N} - \frac{9\sqrt{N}}{\sqrt{N-1}}\right) \\ &\leq \frac{5\pi}{8}\sqrt{N}.\end{aligned} \tag{16}$$

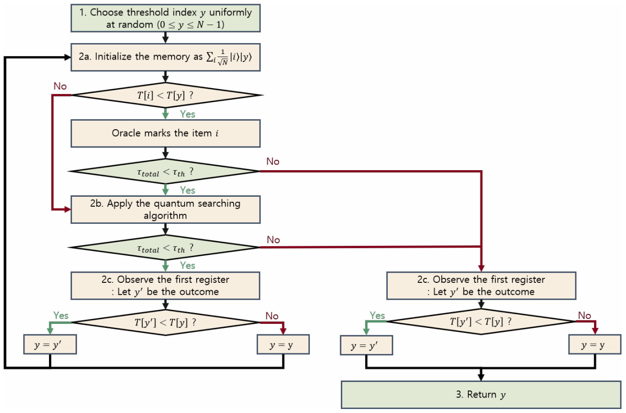

1. Choose threshold index y uniformly at random $(0 \leq y \leq N-1)$
2a. Initialize the memory as $\sum_i \frac{1}{\sqrt{N}}|i\rangle|y\rangle$
No
$T[i] < T[y]$?
Yes
Oracle marks the item i
No
$\tau_{total} < \tau_{th}$?
Yes
2b. Apply the quantum searching algorithm
No
$\tau_{total} < \tau_{th}$?
Yes
2c. Observe the first register : Let y' be the outcome
2c. Observe the first register : Let y' be the outcome
Yes
$T[y'] < T[y]$?
No
$y = y'$
$y = y$
Yes
$T[y'] < T[y]$?
No
$y = y'$
$y = y$
3. Return y

Fig. 2: Quantum Minimum Searching algorithm flow chart.

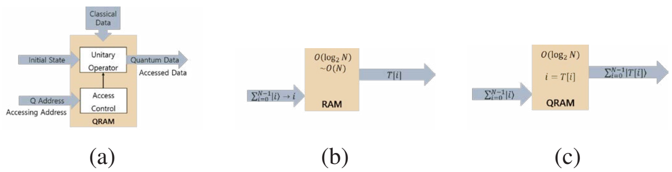
The reason for using (13) is that we will observe the register after all Grover iterations, not during the iterations. Instead of stopping Grover iterations to observe the result and restarting the process described in [4], we observe the register after $O_{qsa}(\sqrt{N/t})$ iterations. Details of the whole process are represented in Section IV.A.

Equation (15) and (16) lead to the fact that the expected total number of iterations before y returns the index of the minimum using the infinite algorithm is at most $m_0 = 0.625\pi\sqrt{N} + 0.7\left(log_2 N\right)^2$ which is demonstrated in [3]. To limit the execution time of the algorithm, we need $2m_0 = 1.25\pi\sqrt{N} + 1.4\left(log_2 N\right)^2$ iterations at most with the probability of success at least $1/2$, unlike $2m_0 = 22.5\sqrt{N} + 1.4\left(log_2 N\right)^2$ in [3]. Thus, the upper bound of $O_{qmsa}(\sqrt{N})$ is expressed as

$$O_{qmsa}\left(\sqrt{N}\right) \leq 2m_0. \tag{17}$$

D. Appplication of QRAM

QRAM(Quantum Random Access Memory) provides an interface between classical data and quantum data. It registers classical data to a massive superposition of entangled quantum states. The best-known model is "*bucket-brigade QRAM*" by Giovannetti in [10]. However, QRAM needs additional cost of loading inputs $O(N)$ which can dominate quantum algorithms' cost. According to the paper [10], bucket-brigade model can reduce the complexity of retrieving data to $O(\log_2 N)$. Upon QRAM model, the cost of the total process can be different. In this paper, we assume that the cost of accessing memory is $O(\log_2 N)$ and it doesn't affect



(a)

(b)

(c)

Fig. 3: QRAM structure and assumption, (a) QRAM structure, (b) The relation of table $T[i]$ to index i in classical RAM, (c) The relation of table $T[i]$ to index i in QRAM model.

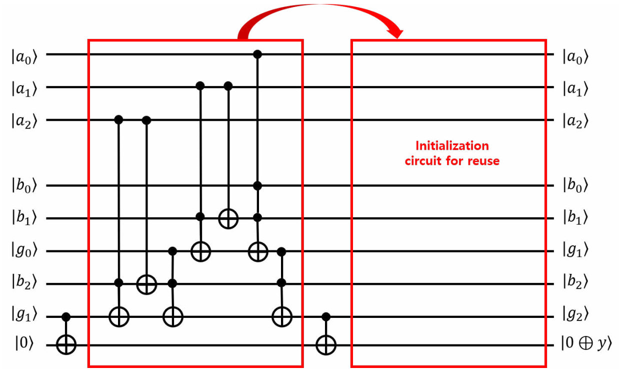


Fig. 4: Circuit of comparator, $n = 3$.

$O_{qmsa}(\sqrt{N})$. In the meantime, QRAM maps the item $T[i]$ to entangled quantum state and makes it possible to apply quantum operators, i.e., quantum comparator described in the next section, to $T[i]$.

III. Circuit

A. Quantum comparator

The constant adder in [6] computes $r = c + b$ where c is a constant and b is a variable. The adder uses dirty qubits g and returns a carry bit of r, which is MSB of r. To design the quantum constant adder, the CNOT gate and NOT gate are removed depending on the constant to be added. Instead of removing gates, we change the constant c to variable a and add a control qubit to each gate. To compare a and b, we compute $a - b$ which can be converted to $a + b'$, where b' is the 1's complement of b. Let the last bit of $a - b$ be y, then 1-bit binary value y is expressed as

$$y = \begin{cases} 1 & \text{if } a > b \\ 0 & \text{otherwise} \end{cases}, \tag{18}$$

where the n-bit binary representations of a and b are $a = (a_{n-1}a_{n-2}\ldots a_0)_2$ and $b = (b_{n-1}b_{n-2}\ldots b_0)_2$. The circuit in Fig. 4 shows a quantum comparator for 3-bit vectors.

B. 3-qubit Comparator and simulation results

We simulate 3-qubit comparator using python library 'qiskit' by IBM Q. Fig.5 shows a schematic of 3-qubit comparator from qiskit library, and Fig.6 shows the histogram of the measurement results when $a = (101)_2 = 5_{10}$. In Fig. 5, each qubit corresponds as follows :

$$a = (q_2 q_1 q_0)_2\,,\ b = (q_6 q_4 q_3)_2\,,\ g = (q_7 q_5)\,,\ y = q_8. \tag{19}$$

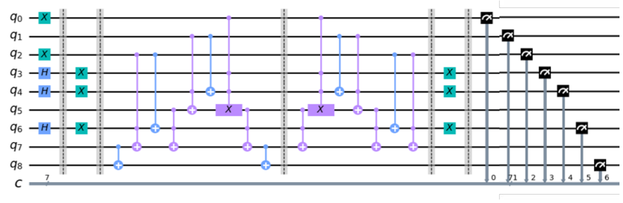


Fig. 5: Schematic of the simulation, $n = 3$.

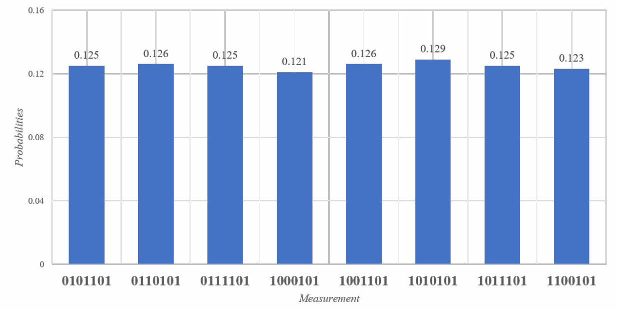


Fig. 6: Probability of each state.

Algorithm 3: Procedure of Algorithm 2.

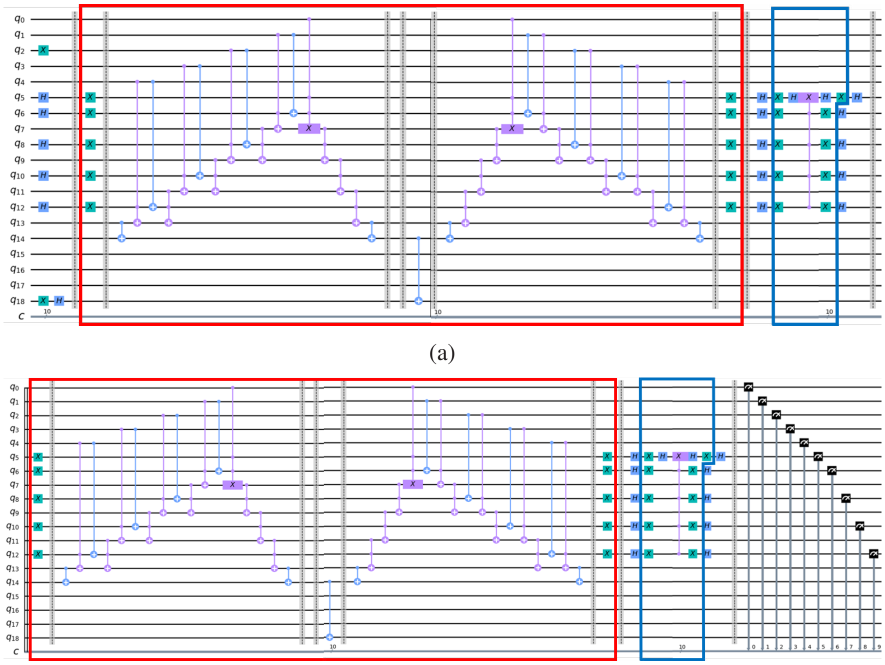
```
Choose threshold index $y$ uniformly at random ($0 \le y \le N-1$).
$\tau_{total} \leftarrow 0$
repeat
    Initialize $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle$ .
    $\tau_{total} \leftarrow \tau_{total} + \log_2 N$.
    repeat
        if $T[i] < T[y]$ then
            Comparator marks the items $i$.
        end if
        Invert a phase using an oracle function, $U_f$.
        Initialize the comparator circuit.
        Amplify the state below the average amplitude
        using Grover operator, $U_G$.
    until $O_{qsa}(\sqrt{N/t})$ times.
    $\tau_{total} \leftarrow \tau_{total} + O_{qsa}(\sqrt{N/t})$.
    Measure the state.
    Observe the register and let $y'$ be the outcome.
    if $T[y'] < T[y]$ then
        $y \leftarrow y'$.
    else
        $y \leftarrow y$.
    end if
until $\tau_{total} \ge O_{qmsa}(\sqrt{N})$.: Minimum searching iteration
return $y$.
```

In Fig. 6, starting from MSB, each bit represents y, a, and b in order. All values less than a, i.e., $(000)_2$ $(100)_2$, have 1 for y. Therefore, Fig. 6 shows that MSB is 1 for these values.

IV. Implementation

A. Procedure

Algorithm. 3 describes a whole procedure of quantum minimum searching algorithm with appropriate timeouts, which are calculated in Section II.C. To calculate $O_{qsa}(\sqrt{N/t})$, we need t which is unknown. In this paper, the threshold y is approximated to t. After initializing the state $|\psi\rangle$, QRAM maps the state $|i\rangle|y\rangle$ to the state $|T[i]\rangle|T[y]\rangle$ in a superposition. To execute the statement, $T[i] < T[y]$, the quantum com-



(a)

(b)

Fig. 7: Schematic of one minimum searching iteration when initial $y = 4$. Red boxes indicate oracle operations and blue boxes indicate Grover operators. (a) the first Grover iteration (b) the second Grover iteration and measurement.

parator described in Section III.A can be applied to the state $|T[i]\rangle |T[y]\rangle$.

B. Circuit : 5-qubit minimum searching algorithm

The circuit needs 5 qubits for each y and i to be compared. Additional 4 qubits for garbage g, ancilla qubits for multiple controlled CNOT gates, and 1 qubit representing the comparator's result are also required. Fig. 7 shows a schematic of one minimum searching iteration when the initial threshold y for a 5-qubit data space is 4. In this case, two Grover iterations is required.

C. Simulation result

Using the schematic in Fig.7 and Algorithm.3, we simulated 5-qubit minimum searching algorithm for items $T[i] = i$ where $i = 0\ N-1$. Table I shows the result of 500 trials with different initial thresholds. The distribution of the minimum over 500 trials is shown in the histogram in Fig. 8. In Table I, the failure rate is 0.8%, which means that the algorithm detects the minimum with a probability of success over 99%. The expected number of iterations $O_{qmsa}(\sqrt{N})$ is 57 using (17). Compared to the time complexity of the classical algorithm which is $O(N)$ or $O(N^2) = N(N+1)/2 = 528$, in the worst case, the algorithm can achieve exponential speedup over classical algorithms.

V. Conclusion

This paper has covered the quantum minimum searching algorithm starting from Grover's searching algorithm. This research aimed to find the minimum among N items in a superposition of entangled quantum states using the quantum minimum searching algorithm. Thus, we have proposed three things to modify the quantum minimum searching algorithm and implement it in 5-qubit space.

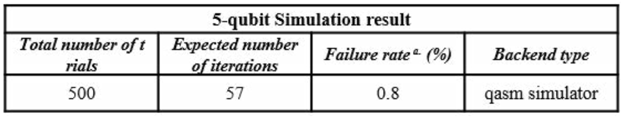
a) Modifying the expected total number of iterations: We calculated the expected total number of iterations $2m_0$ using (13) because we observe the register after Grover iterations. Therefore, $2m_0$ is reduced by 64.8% from 162 to 57 when n=5. Simultaneously, the algorithm achieves high accuracy, such as 99%, shown in Table I.

b) Applying the concept of QRAM: The central idea of QRAM is to register classical data to a massive superposition of entangled quantum states. If we register item $T[i]$ to the quantum state, we can apply quantum operators to superpositioned $T[i]$. Therefore, the speedup from quantum parallelism is expected.

c) Comparing data using quantum comparator: The only thing we need from a comparator is whether b is less than a or not. We can get the information from a carry bit after calculating $a + b'$. Therefore, the quantum comparator using a carry bit is suitable in this case.

The suggestions of this research reduced the expected

TABLE I: 5-QUBIT SIMULATION RESULT



5-qubit Simulation result			
Total number of t rials	*Expected number of iterations*	*Failure rate* [a] *(%)*	*Backend type*
500	57	0.8	qasm simulator

[a] The number of $min(T[i]) = 0$/Total number of simulations)*100

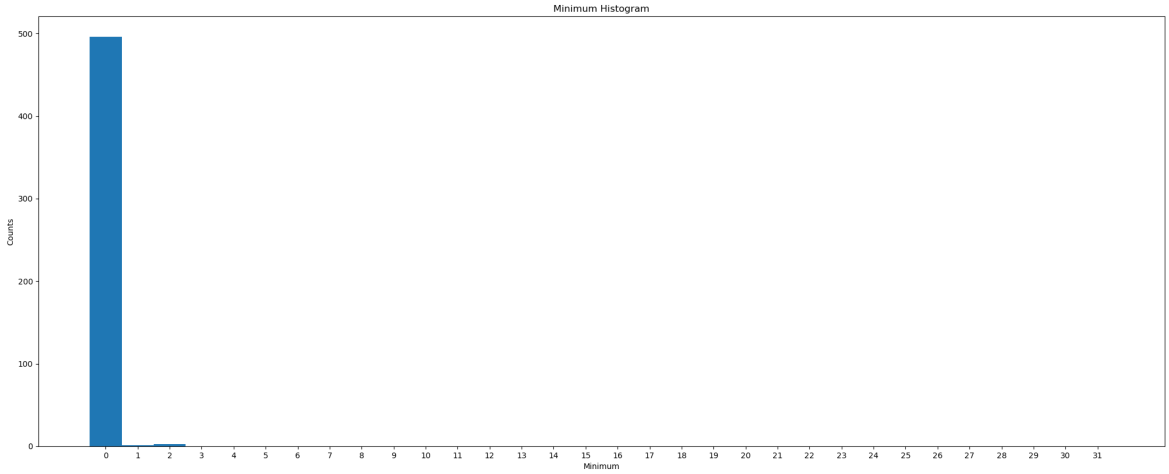


Fig. 8: Histogram of the minimum in 500 experiments.

TABLE II: SIMULATION ENVIORNMENT

Specification	Pycharm	Python	Qiskit library	
	Pycharm community	***Python***	***qiskit***	***qiskit-aer***
Version	2019.3.4	3.6	0.19.6	0.5.2

Specification	Qiskit library				
	qiskit -aqua	***qiskit-aqua -interfaces***	***qiskit-ibmq -provider***	***qiskit -ignis***	***qiskit -terra***
Version	0.6.6	0.2.1	0.7.2	0.3.3	0.14.2

number of iterations while maintaining high accuracy, even though extra time from QRAM and the quantum comparator were $O(\log_2 N)$ each. Meanwhile, whether we guarantee the quality of QRAM operation was not addressed because it was beyond the scope of this research, which has been researched actively. Calculating $O_{qsa}(\sqrt{N/t})$ requires extra operations such as quantum counting to estimate the number of solution t. Because of this limitation in our research, further research will address the method to compensate for it. Besides, it is necessary to analyze the resources of the quantum comparator to estimate the total complexity accurately.

APPENDIX A.

Table II provides the specifications of the environment and library used in this research.

ACKNOWLEDGMENT

This research was supported by Korea Institute of Science and Technology Information(KISTI).

REFERENCES

[1] Peter Wittek. “Quantum Machine Learning”, Elsevier, 2014

[2] M. A. Nielsen and L. I. Chuang, Quantum Computation and Quantum Information 10th Anniversary Edition, Cambridge University Press, 2010.

[3] Christoph Dürr , Peter Høyer , “A quantum algorithm for finding the minimum”, arXiv:quant-ph/9607014, July. 1996.

[4] Michel Boyer, Gilles Brassard, Peter Høyer, “Tight bounds on quantum searching”, PhysComp96, May. 1996.

[5] Lov K. Grover, “A fast quantum mechanical algorithm for database search”, Proc. 28th Ann. ACM Symp. on Theory of Comput., pp. 212 –219, 1996.

[6] Thomas Haner, Martin Roetteler, Krysta M. Svore“Factoring using 2n+2 qubits with toffoli based modular multiplication”, arXiv:1611.07995v2, Jun. 2017.

[7] John Hayes, Igor L. Markov, “Quantum Approaches to Logic Circuit Synthesis and Testing”, AFRL-IF-RS-TR-2006-216, Jun. 2006.

[8] Yanhu Chen, Shijie Wei, Xiong Gao, Cen Wang, Jian Wu , Hongxiang Guo, “An Optimized Quantum Maximum or Minimum Searching Algorithm and its Circuit”, arXiv:1908.07943, Aug. 2019.

[9] Luis Antonio Brasil Kowada, Carlile Lavor, Renato Portugal, Celina M. H. Figueiredo, “A New Quantum Algorithm for Solving the Minimum Searching Problem”, International Journal of Quantum Information, vol. 6, no. 3, pp. 427–436, Mar. 2008.

[10] Vittorio Giovannetti, Seth Lloyd, Lorenzo Maccone, “Quantum Random Access Memory”, PRL 100, Apr. 2008.