ECE523 Final Project Report –
Element Distinctness by Quantum Walk Algorithm

Yuqi Yun, Ming-Tso Wei

April 2017

1 Introduction

Element distinctness problem is determining whether all the elements in a set with N elements are all distinct. In other words, the goal is to find a subset with two or more equal elements. Generally, a set may have two or more subsets with different numbers of equal elements. The case of only one subset of k equal elements is called *element k-distinctness*. For simplicity, we only consider the special case of a single subset with two equal elements ($k = 2$).

Classically, element distinctness can be solved by sorting with $\Omega(N)$ queries. Burhman *et al.* [3] first shows a quantum algorithm with $O(N^{3/4})$ queries. Aaronson and Shi [1] also mathematically prove that the quantum lower bound of element distinctness is $O(N^{2/3})$. In this report, we are focusing on the quantum walk algorithm for element distinctness proposed by Aaronson [2], which only takes $O(N^{2/3})$ queries.

2 The Algorithm

2.1 Quantum Walk Algorithm

The classical counterpart of quantum walk is random walk, in which case each vertex on a graph moves to one of neighboring vertex randomly (for instance, decided by coin flips). The superposition of quantum states naturally provides the randomness for this behavior. Strikingly, it has been shown that quantum walk can be exponentially faster than the classical random walk on some graph with certain complexity. [2]

2.2 Application to Element Distinctiveness

Aaronson claims that if the number of register is limited to $r \leq N^{2/3}$, the run time for solving element distinctness problem can be $O(N^{2/3})$. We define a graph G with $\binom{n}{r} + \binom{n}{r+1}$ vertices to allow quantum walks. We will visually illustrate how quantum walk works in the following section.

2.3 Simulation of Algorithm for k=2

2.3.1 Simple case of N=4

To illustrate the algorithm, we consider the case of $N = 4$ and $x = (x_A, x_B, x_C, x_B)$ with $x_B = x_C$. Then $r = 2$ and there are $\binom{n}{r} = 6$ possible set S with $S \subseteq [N]$ and $|S| = r = 2$, which are {A, B}, {A, C}, {A, D}, {B, C}, {B, D} and {A, D}. There are $\binom{n}{r+1}$ possible set T with $T \subseteq [N]$ and

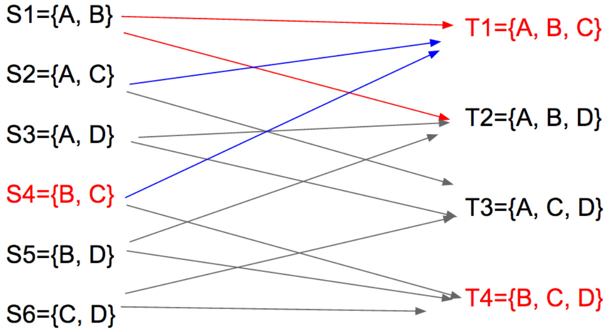

Figure 1: Example of a graph for N=4, k=2

$|T| = r + 1 = 3$, which are {A, B, C}, {A, C, D}, {A, B, D} and {B, C, D}. Then ten vertices are connected shown in Figure 1.

A quantum walk starts from a superposition of sets S's and goes to a superposition of some sets T's. Then from each set T, it goes to a new superposition of set S. The marked vertices in this case is {B, C}. To implement this algorithm, we first firm the $|S, y >$ basis with $y \notin S$, which is consisted of 12 states. (Each of the 6 T's has two choices of y). And we categorized them into $2k + 1 = 5$ types. Let $|\phi_{j,l} >$ be defined if $|S \cap B, C| = j$ and $|y \cap B, C| = l$. Then

$|\phi_{0,0} >= 0$

$|\phi_{0,1} >= \frac{1}{\sqrt{2}}|AD, B > + \frac{1}{\sqrt{2}}|AD, C >$

$|\phi_{1,0} >= \frac{1}{2}|AB, D > + \frac{1}{2}|AC, D > + \frac{1}{2}|BD, A > + \frac{1}{2}|CD, A >$

$|\phi_{1,1} >= \frac{1}{2}|AB, C > + \frac{1}{2}|AC, B > + \frac{1}{2}|BD, C > + \frac{1}{2}|CD, B >$

$|\phi_{2,0} >= \frac{1}{\sqrt{2}}|BC, A > + \frac{1}{\sqrt{2}}|BC, D >$

First, note that all $|S, y >$'s that are marked in the answer are contained in $|\phi_{2,0} >$ only. For amplification of the answers, we flip the sign: $|\phi_{2,0} > \rightarrow -|\phi_{2,0} >$. It can be done by a 5-by-5 matrix U in the $|\phi_{j,l} >$ basis (in the sequence listed above).

$$U = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

Then we can apply the Grover diffusion operator. In this specific case $|S > |y >$ is mapped to $|S > |y' >$ for $y \notin y'$. So $|\phi_{0,1} >$ will be unchanged since $|AD, B >$ will be mapped to $|AD, C >$ and vice versa, but both $|AD, B >$ and $|AD, C >$ are in $|\phi_{0,1} >$ itself. Same for $|\phi_{2,0} >$. To the contrary, $|AB, D >$ will be mapped to $|AB, C >$. Actually, every $|S, y >$ in $|\phi_{0,1} >$ will be mapped to some $|S, y' >$ in $|\phi_{1,0} >$ and vise versa.So the matrix U1 for the transformation in the $|\phi_{j,l} >$ basis is:

$$U1 = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Similarly, we can figure out the Grover diffusion in Algorithm 1, step 4. which is

$$U2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{3} & \frac{2\sqrt{2}}{3} & 0 & 0 \\ 0 & \frac{2\sqrt{2}}{3} & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{2\sqrt{2}}{3} \\ 0 & 0 & 0 & \frac{2\sqrt{2}}{3} & -\frac{1}{3} \end{bmatrix}$$

We initialized $|\phi_{start} >$ to be the uniform superposition of all $|S, y >$'s, then in $|\phi_{j,l} >$ basis,

$|\phi_{start} >= (0, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{6}})$. Then since $t_1 = (\frac{N}{r})^{k/2}$ and $t_2 = \frac{\pi\sqrt{r}}{3\sqrt{k}} + 1$ in this case, the whole algorithm is to apply $((U_2U_1)^{t_2}U)^{t1}$ to $|\phi_{start} >$. Define p as the vector for the probability of the measurement for each $|\phi_{j,l} >$, i.e.

$p = (P(|\phi_{0,0} >), P(|\phi_{0,1} >), P(|\phi_{1,0} >), P(|\phi_{1,1} >), P(|\phi_{2,0} >))$

Then for the N=4 case, the result is $p = (0, 0.229259, 0.603617, 0.142712, 0.0244119)$. Note that the correct result should have $P(|\phi_{2,0} >) > 0.5$ as a sign that we find a marked vertex. We found that if the runtime t_2 for quantum walks is reduced to 1, then the result can be corrected by $p = (0.100823, 0.102881, 0.201646, 0.59465)$. This correction is also utilized in our circuit implementation in Section 3. We will show in the next section that the correct result will be obtained for $N > 10$ though not in this particular case.

2.4 Element distinctness for arbitrary number of elements

Since for element distinctness, we fix k=2, so no matter how large is N, we can represent U, U1 and U2 as 5×5 matrices in the $|\phi_{j,l} >$ basis.The way to compute U, U1 and U2 is specified by the paper. Here I will just illustrate how to write $|\phi_{start} >$ in terms of $|\phi_{j,l} >$'s in general.

$< \phi_{j,0}|\phi_{start} >= \frac{\sqrt{\binom{k}{j}\binom{N-k}{r-j}((N-k)-(r-j))}}{\sqrt{\binom{N}{r}(N-r)}}$

$< \phi_{j,1}|\phi_{start} >= \frac{\sqrt{\binom{k}{j}\binom{N-k}{r-j}(k-j))}}{\sqrt{\binom{N}{r}(N-r)}}$

Since the total number of possible $|S, y >$ combinations are the product of ways of choosing r elements from all N elements and choosing one element from the rest of N-r elements. The number of possible $|S, y >$ combinations contained in $|\phi_{j,l} >$ is calculated by first choosing j elements from the k elements that have formed a k-collision, and then choosing the rest r-j elements from the n-k distinct elements to form S. Finally, y is selected from the remaining (k-j) elements in the k-collision if l=1, or from the remaining (n-k)-(r-j) distinct elements in if l=0.

We have written a program in Mathematica to simulate the measurement result for arbitrary N (for k=2). (You can change N only without modifying any other parts of the code). The codes can be download on-line from: https://duke.box.com/s/0qzbxnvl1y6p5vnh0bbpa2sd6g55gj5m. Some selected simulation results are listed below:

$N = 15, r = 6, t1 = 2, t2 = 2$

$p = (0.00256282, 0.169971, 0.00734126, 0.087622, 0.732503)$

$N = 500, r = 62, t1 = 8, t2 = 6$

$p = (0.0307232, 0.0496646, 0.0118242, 0.161007, 0.746782)$

$N = 234857, r = 3806, t1 = 61, t2 = 46$

$p = (0.0226436, 0.0108335, 0.000577899, 0.242028, 0.723917)$

In conclusion, the algorithm picks out the correct answer $|\phi_{2,0} >$ for all cases where N¿10.

3 Quantum Circuit Implementation

The quantum walk algorithm can be described by a sequence of unitary transformations. Note that the unitary matrices, such as U, U1 and U2, shown in Section 2.3, are block-diagonal with 2x2 matrices, which means that they can be converted into single-qubit gates.

In this section, the circuit implementation utilizes IBM Quantum Experience 2.0 (url). Due to the limitation of about 100 gates, the implemenation is only done with a simple case of N=4 and k=2. In which case, the first componet of the initial state $|\phi_{0,0} >$ is just 0 and not altered by any of the unitary transformations. Therefore, the unitary matrices can be reduced to 4x4 matrices and represented by only two qubits.

The first unitary transformation U, which does the conditional phase flip, is just a controlled-Z gate, as shown below: Figure 2.

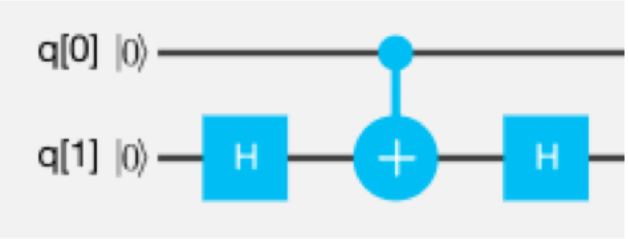

Figure 2: Quantum circuit of conditional phase flip U

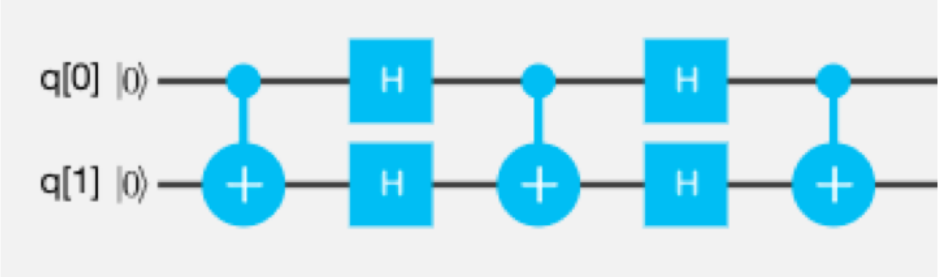

Figure 3: Quantum circuit of the first step of a quantum walk U1

The next unitary transformation U1, the first step of quantum walk, is just a SWAP gate, as shown in Figure **??** For the second step of quantum walk, the unitary matrix U2 is actually block-diagonal; therefore, it can be represented as two controlled U gates, one corresponding to the topleft 2×2 unitary while the other corresponding to the bottom left 2×2 unitary. Since IBM Quantum Experience does not directly provide the implementation of a controlled-U gate, each controlled U gate needs to be decomposed into CNOTs and single-qubit gates $U = e^{i\alpha}AXBXC$, where A, B, C are single-qubit matrices (See Figure 4.6 in Ref. [4]). The resulting quantum circuit is shown in Figure 4.

As the circuits of the subroutines U, $U1$ and $U2$ are constructed, the quantum walk algorithm can be realized by matrix multiplications $((U2U1)^{t_2}U)^{t_1}$, which is just $U_2U_1UU_2U_1U$ in this special case $(N = 4, k = 2)$. For simplicity, the initial state is prepared by Hadamard gates, instead of $|\phi_{start} >$ since the resulting probabilities of the final state are almost the same for both types of initial states.

The simulation results and the real execution results with IBM quantum computers are summarized in Table 1. The final state $|\phi_{2,0} >$ has the highest probability (about 0.6). Both the IBM

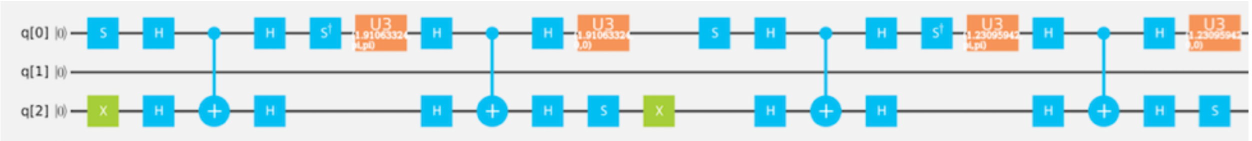

Figure 4: Quantum circuit of the second step of a quantum walk U2

	$P(\|\phi_{0,1}>)$	$P(\|\phi_{1,0}>)$	$P(\|\phi_{1,1}>)$	$P(\|\phi_{2,0}>)$
Numerical (Mathematica)	0.100823	0.102881	0.201646	0.59465
IBM Simulator (100 shots)	0.210	0.010	0.220	0.560
IBM Simulator (8192 shots)	0.148	0.031	0.155	0.666
IBM real processor (1024 shots)	0.176	0.093	0.231	0.500
IBM real processor (8192 shots)	0.154	0.114	0.239	0.493

Table 1: Comparisons of numerical simulation, simulation done by IBM quantum simulator and the execution results from the IBM real processors

simulators and real execution results agree well with the numerical results we obtained.

4 Discussion and Conclusion

The two-qubit results obtained by the IBM quantum computer is limited in several aspects. First, both the IBM quantum computer and simulator limit the number of gates (about a hundred at most). With this limitation, more generalized cases for $N > 4$ would be not accessible since the number of gates will also grow with $N : t_1 = (N/r)^{k/2}$. Moreover, in Section 2, we have shown that the algorithm is actually more accurate when N is large. As a result, the performance of the algorithm is sacrificed by the limited number of gates as well.

In addition, the quantum walk algorithm for element distinctness ($k = 2$) does not require more than three qubits. The generalization to element k-distinctness, which requires $(2k+1) \times (2k+1)$ matrices would significantly increase the complexity of the problem. All in all, according to our numerical simulation, the quantum walk algorithm for element distinctness should work properly as long as the quantum computation resource is not limited.

References

[1] S. Aaronson and Y. Shi, "Quantum lower bounds for the collision and the element distinctness problems", J. ACM **51**, 595–605 (2004).

[2] A. Ambainis, "Quantum walks and their algorithmic applications", Int. J. Quantum Inf. **01**, 507–518 (2003).

[3] H. Buhrman, C. Dürr, M. Heiligman, P. Høyer, F. Magniez, M. Santha, and R. de Wolf, "Quantum algorithms for element distinctness", SIAM J. Comput. **34**, 1324–1330 (2005).

[4] M. Nielsen and I. Chuang, *Quantum computation and quantum information*, 10th Anniversity Edition (Cambridge University Press, 2010).