# Quantum Walks and their Applications

Shruti Agrawal
14668

Kshiitiz Singhal
14330

**Advisor**: Prof. Rajat Mittal
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

November 2017

**Abstract**

Random walks are best described as stochastic processes, where a 'walker' may take steps in a mathematical space in different directions with some random probability distribution. In this project we firstly gathered an initial understanding of the Quantum analogue of these random walks. After that we studied a 2003 paper by Andris Ambainis, Quantum Walk algorithm for Element Distinctness, which provides a $\mathcal{O}(N^{2/3})$ query complexity algorithm for the problem, compared to the $\mathcal{O}(N)$ queries classical algorithm.

# Contents

# 1 Introduction

Random walks are processes where an object moves in a mathematical space, taking 'steps' in different directions with certain probabilities. Given the randomness that lies within the Quantum world, Quantum Computing serves as a vital tool in solving such problems. Interesting problems that can be solved using these quantum tools at our hand is the Element Distinctness Problem[4] and L-Subset Finding Problem[6], which finds applications in modern world networking solutions. In this project, we studied at depth both of these problems.

# 2 Quantum Walk on a Line

In the process of studying Quantum Walks, let us first start by getting an understanding of their classical counterpart. We shall firstly look at the simplest of the random walks, i.e. walk on a straight line. For the sake of simplicity and symmetry we will start at position 0 and move either left or right with an equal probability of $\frac{1}{2}$. We know that such a walk yields a normal probability distribution, with a peak at 0.

The Quantum Walk on a straight line is different in certain aspects. Unlike the classical, where the particle is present at any single location, the Quantum particle is present at a superposition of states. Now let us define our Hilbert Space for the walk. We consider $\mathcal{H_P}$ to be our position Hilbert space, denoting the positions of the particle. The basis states are $\{|i\rangle : i \in \mathbb{Z}\}$.

We also augment a new Hilbert Space, $\mathcal{H_C}$, called the coin space, used to denote the direction of movement. This is spanned by two basis states, $\{|left\rangle, |right\rangle\}$. We will look at the need of this extra space later. Hence our states in the complete system are in $\mathcal{H_C} \otimes \mathcal{H_P}$.

Now let us define our unitary operators for performing the walk. The first unitary is the conditional shift operator given as:

$$S = |right\rangle \langle right| \otimes \sum_i |i+1\rangle \langle i| + |left\rangle \langle left| \otimes \sum_i |i-1\rangle \langle i|$$

This transforms the basis states $|left\rangle \otimes |i\rangle$ to $|left\rangle \otimes |i-1\rangle$ and $|right\rangle \otimes |i\rangle$ to $|right\rangle \otimes |i+1\rangle$.

The second operator is a rotation in the coin space, also called a coin flip. One such 'balanced' coin is the Hadamard coin, defined as:

$$|right\rangle \otimes |0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|left\rangle + |right\rangle) \otimes |0\rangle$$

$$|left\rangle \otimes |0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|right\rangle - |left\rangle) \otimes |0\rangle$$

A quantum walk step includes a coin flip, followed by the bit operator. So if we start in state $|right\rangle \otimes |0\rangle$, after one step of the walk(SH) we obtain the state, $\frac{1}{\sqrt{2}}(|left\rangle \otimes |-1\rangle + |right\rangle \otimes |1\rangle)$. The quantum walk of t steps is given by $U^t$ where U is given by $U = S.(C \otimes I)$. We can start our walk in the state $|right\rangle \otimes |0\rangle$.

## 2.1 Need of the Coin Space

Intuitively, in a quantum walk on a line we should move to superposition of neighbouring states in each step, and the operation should look like,

$$|i\rangle \to a\,|i+1\rangle + b\,|i-1\rangle$$

At first glance, the position space alone might seem sufficient for the walk. One might even be tempted to define the shift operator as:

$$S = \frac{1}{\sqrt{2}} \sum_i (|i+1\rangle \langle i| + |i-1\rangle \langle i|)$$

If we take a closer look, it easy to see that this operation is not unitary unless, $|a|^2 = 1$ or $|b|^2 = 1$. Then this operation becomes trivial and the same as the classical case. Consider the orthogonal states, $|i\rangle$ and $|i+2\rangle$. On applying the shift operator such states no more remain orthogonal, violating the conditions for a valid operator. Hence we are required to augment an additional coin space, so as to differentiate between such cases.

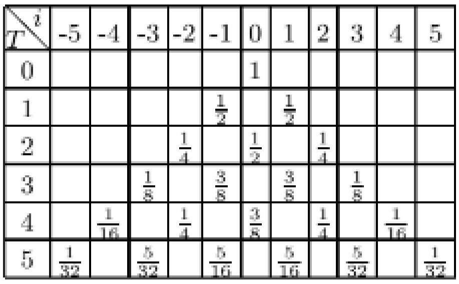
## 2.2 Analysis of the Walk

The first step is a rotation in the coin space i.e a 'coin flip' followed by the conditional shift operator S. So,

$$|right\rangle \otimes |0\rangle \xrightarrow{SH} \frac{1}{\sqrt{2}}(|left\rangle \otimes |-1\rangle + |right\rangle \otimes |1\rangle)$$

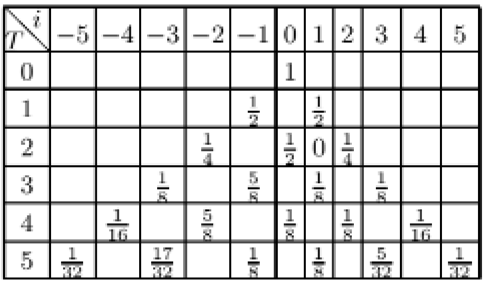
. Thus, The quantum walk of t steps is given by $U^t$ where U is given by $U = S.(C \otimes I)$

Comparing the classical and quantum analogs of random walks on the straight line, we get the following probability distributions.



| $T \backslash i$ | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | | | | | | 1 | | | | | |
| 1 | | | | | $\frac{1}{2}$ | | $\frac{1}{2}$ | | | | |
| 2 | | | | $\frac{1}{4}$ | | $\frac{1}{2}$ | | $\frac{1}{4}$ | | | |
| 3 | | | $\frac{1}{8}$ | | $\frac{3}{8}$ | | $\frac{3}{8}$ | | $\frac{1}{8}$ | | |
| 4 | | $\frac{1}{16}$ | | $\frac{1}{4}$ | | $\frac{3}{8}$ | | $\frac{1}{4}$ | | $\frac{1}{16}$ | |
| 5 | $\frac{1}{32}$ | | $\frac{5}{32}$ | | $\frac{5}{16}$ | | $\frac{5}{16}$ | | $\frac{5}{32}$ | | $\frac{1}{32}$ |

Figure 1: The probability of being at position i after T steps of the classical random walk on the line starting in 0.



| $T \backslash i$ | $-5$ | $-4$ | $-3$ | $-2$ | $-1$ | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | | | | | | 1 | | | | | |
| 1 | | | | | $\frac{1}{2}$ | | $\frac{1}{2}$ | | | | |
| 2 | | | | $\frac{1}{4}$ | | $\frac{1}{2}$ | 0 | $\frac{1}{4}$ | | | |
| 3 | | | $\frac{1}{8}$ | | $\frac{5}{8}$ | | $\frac{1}{8}$ | | $\frac{1}{8}$ | | |
| 4 | | $\frac{1}{16}$ | | $\frac{5}{8}$ | | $\frac{1}{8}$ | | $\frac{1}{8}$ | | $\frac{1}{16}$ | |
| 5 | $\frac{1}{32}$ | | $\frac{17}{32}$ | | $\frac{1}{8}$ | | $\frac{1}{8}$ | | $\frac{5}{32}$ | | $\frac{1}{32}$ |

Figure 2: The probability of being found at position i after T steps of the quantum random walk on the line, with the initial state $|left\rangle \otimes |0\rangle$

The quantum walk begins to differ from the classical case from $T = 3$ onwards. The quantum walk is also asymmetric with a drift towards the left. This asymmetry arises because the hadamard coin we used treats the two directions $|left\rangle$ and $|right\rangle$ differently. There is a phase shift only in the case of $|left\rangle$. The obtained probability distribution looks like,

To get rid of this asymmetry we can do either of the following. We can start in a state which is a superposition of $|left\rangle$ and $|right\rangle$

$$|\phi_{sym}\rangle = \frac{1}{\sqrt{2}}(|left\rangle + \iota\,|right\rangle) \otimes |0\rangle$$

Or we can use a balanced coin instead that treats the two directions similarly.

$$Y = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & \iota \\ \iota & 1 \end{bmatrix}$$

The obtained symmetric distribution looks like,

Some important differences between classical and quantum random walks are noted to be.

- The classical random walk has max. probability for $n = 0$, whereas in the quantum case the maximum probability is reached for $|n| \approx \pm\frac{T}{\sqrt{2}}$
- A classical walk on a line has a variance $\sigma^2 = T^2$ thus the expected distance from the origin is of $\theta(T)$. In the quantum case the variance scales as $\sigma^2 \sim T^2$. The expected distance from the origin is of the order $\sigma \sim T$. The quantum walk propagates quadratically faster.

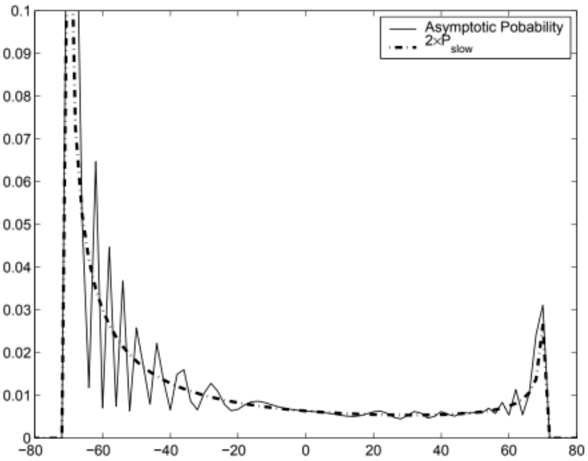


Figure 3: The probability distribution of the quantum random walk on the line, with the initial state $|left\rangle \otimes |0\rangle$ after $T = 100$ steps
. Only the probabilities at even points is plotted since the probabilities at odd points is zero.

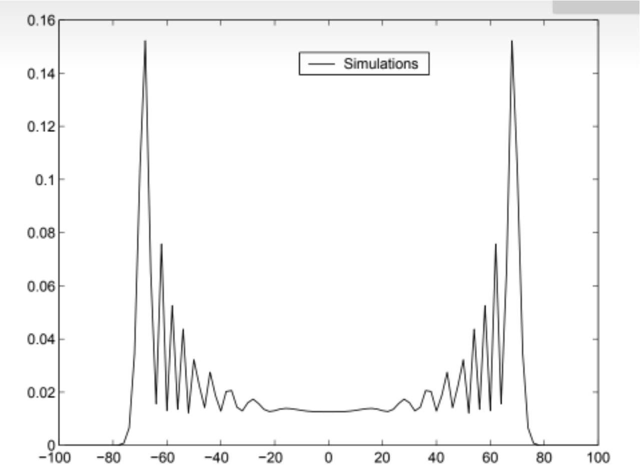


Figure 4: The probability distribution of the quantum random walk on the line, with the symmetric initial state for $T = 100$ steps. Only the probabilities at even points is plotted since the probabilities at odd points is zero.

These properties are exploited to design fast quantum algorithms. Another interesting quantum effect appears by adding boundary conditions to the line.

## 2.3 Bounded walk on Line. Classical vs. Quantum

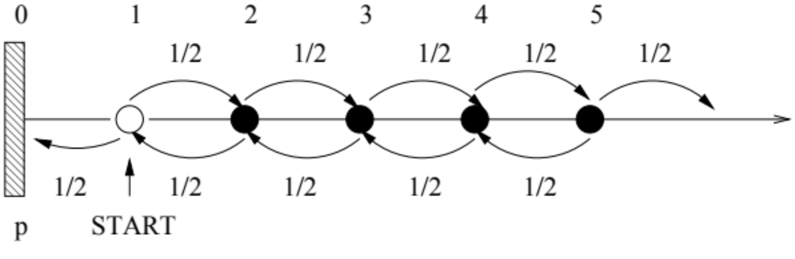


Figure 5: Classical random walk on a bounded line with an absorbing boundary at position 0 and the starting position 1.

An absorbing boundary at position $|b\rangle$ corresponds to a partial measurement at every step. One step of the walk now becomes U followed by $M_b$ on $\mathcal{H_P}$ described by the two projectors onto $|b\rangle$ and $B_\perp$

$$M_b\,|\psi\rangle = \begin{cases} |b\rangle & p_b = |\,\langle b|\psi\rangle\,|^2 \\ \frac{|\psi\rangle - \langle b|\psi\rangle |b\rangle}{\sqrt{1-|\langle b|\psi\rangle|^2}} & p_{b_\perp} = 1 - |\,\langle b|\psi\rangle\,|^2 \end{cases}$$

If the measurement result of $M_b$ gives $|b\rangle$ then the walk is stopped, else the next iteration is carried out.

In the classical case the probability to ever get absorbed at the origin is $p = 1$. It can be seen via a recursive reasoning, where $p_{20}$ is the probability of ever hitting 0 from 2, which is equal to $p_{21} * p_{10}$ (the probability of ever hitting 1 from 2 and the probability of ever hitting 0 from 1 ). Since both $p_{21}$ and $p_{10}$ are equal to $p$ we have,

$$p = \frac{1}{2} + \frac{1}{2}p_{21}p_{10} = \frac{1}{2}(1 + p^2)$$

This has the solution $p = 1$. Thus in the classical case a random walk always gets absorbed by a boundary and never escapes to infinity.

For a Quantum Hadamard random walk with an absorbing boundary at $|0\rangle$ starting in the state $|left\rangle \otimes |1\rangle$ there is finite non zero escape probability. The escape probability is non zero and finite for different initial states and distances to the boundary.

## 2.4 Quantum Walk on general graphs

Let us consider the case of d-regular graphs first, where the coin space $\mathcal{H_C}$ has dimension d. For every vertex we label it's outgoing edges $j \in 1...d$ and an edge $e = (j, w)$ is denoted by $e_v^j$. State for a vertex v, pointing along edge j is $|j\rangle \otimes |v\rangle$. The conditional shift operator S is defined as -

$$S\,|j\rangle \otimes |v\rangle = \begin{cases} |j\rangle \otimes |w\rangle & e_v^j = (v, w) \\ 0 & otherwise \end{cases}$$

S is defined similarly even if graph is not d-regular. The coin flip operator is a d-dimensional unitary transformation. In case we want an equi-probable coin, we can Direct Fourier Transform(DFT). The Grover Diffusion operator used in the Grover's Search algorithm[3] can also be used as a coin operator.

# 3 Element Distinctness Problem

The element distinctness problem is the problem of determining whether all the terms in a list are distinct or not. The formal definition of the problem can be given as:
Element Distinctness: Given numbers $x_1, x_2..., x_N \in$ [M], are there i,j $\in$ [N], i $\neq$ j such that $x_i = x_j$?
Element k-Distinctness: Given numbers $x_1, x_2..., x_N \in$ [M], are there k distinct indices $i_1, i_2, ..., i_k \in$ [N], such that $x_{i_1} = x_{i_2} = .... = x_{i_k}$?
We will be considered with the query complexity of the problem rather than the time complexity. Our oracle takes in input i, and outputs corresponding $x_i$. Classically, the problem requires $\Theta(N)$ queries even for a randomized algorithm. In 2003, Andris Ambainis[4] gave an $\Theta(N^{2/3})$ query complexity quantum algorithm for this problem using random walks. More generally, the algorithm solves the Element k-distinctness problem in $\Theta(N^{k/k+1})$ queries.

## 3.1 Generalized Problem: L-Subset Finding

In the same year, Andrew Childs and Jason Eisenberg[6] showed that the algorithm given by Ambainis solved a much more general problem of L-subset finding. The input here consists of a black box function $f : D \rightarrow R$ such that D($|D| = N$) and R are finite and the property $P \subset (D \times R)^L$. The output is some L-subset $\{x_1, x_2....x_l\}$ s.t. $((x_1, f(x_1)), (x_2, f(x_2))....(x_L, f(x_L))) \in P$. For $L = 1$, this is the same as the unstructured search problem, solved using Grover's Search Algorithm[3]. For $L = 2$, this is a case of our element distinctness problem with $P = \{((x_1, y), (x_2, y))\}$.

## 3.2 Algorithm

### 3.2.1 Hilbert Space

To define our algorithm, we once again look at our Hilbert Space first. Our graph consists of vertices formed by all subsets of domain D, of size either M or M+1. Let A $\subset$ D with $|A|$ = M and B $\subset$ D with $|B|$ = M+1. Vertices A and B are connected iff A $\subset$ B, i.e. for some k $\in$ $D\backslash$A, B = A $\cup\{k\}$.
We now consider an orthonormal basis of our state $|A\rangle$, one for each subset. For each subset A with $|A|$ = M, there are M associated functions values f(A) $\in R^M$ and similarly for B. These values can be obtained using our input oracle and thereafter we store them along with each subset. As we move from vertex to vertex we will be required to update these function values too, which would again require queries to the oracle.
We define our complete state on which we compute as $|A, f(A), k\rangle$, where $|A, f(A)\rangle$ is as described above and $|k\rangle$ is our coin register in this case, where k $\in$ D. If $|A|$ = M, then k indicates an element to be added to A, so we have k $\in$ $D\backslash$A and for $|B|$ = M+1, k indicates an element that can be removed from B, so we have k $\in B$.

### 3.2.2 Transformation Operators

A single step of our walk, W can be described as a product of four unitary transformations, $W = SC_2SC_1$. The shift operator S acts as,

$$S\,|A, f(A), k\rangle = |A \cup \{k\}, f(A \cup \{k\}), k\rangle \tag{1}$$

$$S\,|B, f(B), k\rangle = |B \setminus \{k\}, f(B \setminus \{k\}), k\rangle \tag{2}$$

and can be implemented using one query to the input oracle. The coin operators $C_1$ and $C_2$ are Grover diffusion operators on $k \notin A$ and $k \in B$ respectively.

$$C_1\,|A, f(A), k\rangle = |A, f(A), k\rangle - \frac{2}{N - M} \sum_{k` \notin A} |A, f(A), k`\rangle \tag{3}$$

$$C_2\,|B, f(B), k\rangle = |B, f(B), k\rangle - \frac{2}{M + 1} \sum_{k` \in B} |B, f(B), k`\rangle \tag{4}$$

$C_1$ and $C_2$ act as identity operator on all other states not included here. Notice that these do not require any queries to the oracle. Along with this we also require a phase flip operation that distinguishes subsets A that include an L-subset satisfying $\mathcal{P}$. $P|A, f(A)\rangle = -|A, f(A)\rangle$ if $S \subset A$ and $|A, f(A)\rangle$ if $S \not\subset A$. Implementing P too does not require any additional queries as all the values required to calculate it are already there with us.

### 3.2.3 Complete Algorithm

The initial state is,

$$|s\rangle = \frac{1}{\sqrt{c}} \sum_{|A|=m} |A, f(A)\rangle \sum_{k \notin A} |k\rangle \tag{5}$$

where c is $\binom{N}{M}(N-M)$. Creating such a state requires M queries to the blackbox.

The full algorithm can be written as, $(W^{t_1}P)^{t_2}$. Hence our output state would be $(W^{t_1}P)^{t_2}|s\rangle$. As each step of our walk requires 2 queries, the total queries required are $M + 2t_1t_2$.

## 3.3 Analysis of the Algorithm

Based on the eigenvalues, we take $t_1 = \lfloor \frac{\pi}{2}\sqrt{M/L} \rceil$ and $t_2$ is calculated to be $\lfloor \frac{\pi}{2}(N/M)^{L/2} \rceil$. Also, choosing $M = \lfloor N^{\frac{L}{L+1}} \rceil$ the query complexity $M + 2t_1t_2$ becomes $\Theta(N^{\frac{L}{L+1}})$, our desired complexity. Finally if we consider our desired state to be,

$$|w\rangle = \frac{1}{const} \sum_{|A|=M, |A\cap S|=L} |A, f(A)\rangle \sum_{k \notin A, k \notin S} |k\rangle \tag{6}$$

then,

$$|\langle w| (W^{t_1}P)^{t_2} |s\rangle|^2 = 1 - \Theta(1/M + M/N) \tag{7}$$

Therefore, a measurement of the state $(W^{t_1}P)^{t_2}|s\rangle$ yields a subset A for which $S \subset A$, together with the associated function values f(A), with probability close to 1.

## 3.4 Proof of the Algorithm

To look into the proof of the algorithm, let us firstly divide our Hilbert Space into (2k+1)-dimensional subspaces. Let $S_1 = S$ and $S_0 = D \setminus S$. The states for the subspace are such:

$$|A_{j,p}\rangle = \frac{1}{\sqrt{c_{j,p}}} \sum_{|A|=M, |A\cap S|=j} |A, f(A)\rangle \sum_{k \notin A, k \in S_p} |k\rangle$$

for $j = 0, 1, ..., L-1, p = 0, 1$ and for $j = L, p = 0$ and the states,

$$|B_{j,p}\rangle = \frac{1}{\sqrt{d_{j,p}}} \sum_{|B|=M+1, |A\cap S|=j} |B, f(B)\rangle \sum_{k \in A, k \in S_p} |k\rangle$$

for $j = 0, p = 0$ and $j = 1, ..., L, p = 0, 1$.

The proof of this algorithm relies on the following two lemmas:

**Lemma 1.** *The eigenvalues of W are 1 and exp* $\left[\pm\iota(2\sqrt{j/M} + \mathcal{O}(1/M + 1/N))\right]$ *for* $j = 1, 2, ...L$. *The eigenvector with eigenvalue 1 is* $|s\rangle$, *and the two eigenvectors with eigenvalues corresponding to* $j = L$ *are* $\frac{1}{\sqrt{2}}(|A_{L-1,1}\rangle \pm \iota |A_{L,0}\rangle) + \mathcal{O}(\sqrt{M/N})$.

**Lemma 2.** $W^{t_1}P$ *has two eigenvectors* $|\theta_\pm\rangle = \frac{1}{\sqrt{2}}(|w\rangle \pm \iota |s\rangle) + \mathcal{O}(1/M + M/N)$, *with eigenvalues exp* $\left[\pm 2\iota(M/N)^{L/2}(1 + \mathcal{O}(1/M + M/N))\right]$.

## 3.5 Co-relation with the Collision Problem

The collision problem is such, given a 2-1 function f, find a pair x, y $\in$[N] such that f(x)= f(y). The classical deterministic algorithm takes total $(N/2+1)$ queries. The randomized algorithm can achieve this $\Theta(N^{1/2})$ queries, similar to the famous birthday problem. In 2001, Yaoyun Shi[5] proved the quantum lower bounds for the element distinctness problem based on the lower bounds of the collision problem proved earlier by Scott Aaronson[7]. This was done by converting instances of Collision Problem into Element Distinctness problem.

We run the algorithm for finding element distinctness on $\Theta(N^{1/2})$ inputs. As the oracle is two-to-one, by the Birthday problem, we know that we are likely to find a collision in these inputs. Hence we can find a collision pair in $\Theta(N^{1/3})$, which had been shown to be the tight lower bound for the collision problem[7].

## 3.6 Finding an L-clique in a N-Vertex Graph

An L-clique is a subset of size L of the N vertices such that their induced subgraph(graph containing all those vertices and the edges joining them to each other) is complete. Our input oracle takes in two vertices and outputs whether there is an edge between them.

For L=2, this is equivalent to finding an edge in a graph, i.e. an unstructured search over $\binom{N}{2}$ edges. For L=3, this is the triangle finding problem whose naive Grover's Search algorithm solves the problem in $\Theta(N^{3/2})$ queries by running over $\binom{N}{3}$ triplets. Application of element distinctness can solve it in $\Theta(N^{4/3})$ queries.

To convert this problem, each subset A(as used in the algorithm) consists of M vertices, where along with them all the edges among these vertices are stored. This would require $\Theta(M^2)$ queries compared to M queries previously. Also now each step of the walk would require $\Theta(M)$ queries to compute all the edges associated with the newly added/removed vertex. Hence the total queries would be $\Theta(M^2 + (N/M)^{L/2} * \sqrt{M} * M)$. For $M = \lfloor N^{L/L+1} \rceil$, we get an overall query complexity of $\Theta(N^{\frac{2L}{L+1}})$.

## 4 Conclusion

In the process of understanding Quantum Walks, we needed to comprehend the classical random walks first. In this we came across, Markov Chains, stationary distributions and mixing time. Thereafter, in an attempt to understand Ambainis's paper, we were led to Child's use of that paper to give a solution to an even bigger problem. We are glad to have chosen such a great topic for our project, which promises solutions to problems that we unknowingly encounter in our everyday lives.

## 5 Acknowledgments

We are thankful to Prof. Rajat Mittal for giving us this opportunity to pursue such a wonderful field in the world of Quantum Computing and the guidance that we continuously received from him. We are also thankful to all the other students who could be a part of this journey, whether in the form of presence during the presentation or any other support rendered. We would also like to express our sincere gratitude to Talla Aravind Reddy, for discussing some of our problems with us.

## References

[1] Julia Kempe (2003) Quantum random walks - an introductory overview. *Contemporary Physics,* 44(4), 307 – 327.

[2] Andris Ambainis (2004) Quantum walks and their algorithmic applications. *International Journal of Quantum Information,* 507 – 518.

[3] Lov K. Grover (1996) A fast quantum mechanical algorithm for database search. *Symposium on the Theory of Computing (STOC)*, 212 – 219.

[4] Andris Ambainis (2003) Quantum walk algorithm for element distinctness. *SIAM Journal on Computing,* 37(1), 210 – 239.

[5] Yaoyun Shi (2001) Quantum lower bounds for the collision and the element distinctness problems.

[6] Andrew Childs and Jason M. Eisenberg (2003) Quantum algorithms for subset finding.

[7] Scott Aaronson (2001) Quantum lower bounds for the collision problem.