Quantum walk algorithm for element distinctness

Andris Ambainis*

Abstract

We use quantum walks to construct a new quantum algorithm for element distinctness and its generalization. For element distinctness (the problem of finding two equal items among N given items), we get an $O(N^{2/3})$ query quantum algorithm. This improves the previous $O(N^{3/4})$ quantum algorithm of Buhrman et al. [10] and matches the lower bound by Shi [28]. We also give an $O(N^{k/(k+1)})$ query quantum algorithm for the generalization of element distinctness in which we have to find k equal items among N items.

1. Introduction

Element distinctness is the following problem.

Element Distinctness. Given numbers $x_1, \ldots, x_N \in [M]$, are they all distinct?

It has been extensively studied both in classical and quantum computing. Classically, the best way to solve element distinctness is by sorting which requires $\Omega(N)$ queries. In quantum setting, Buhrman et al. [10] have constructed a quantum algorithm that uses $O(N^{3/4})$ queries. Shi [28] has shown that any quantum algorithm requires at least $\Omega(N^{2/3})$ quantum queries.

In this paper, we give a new quantum algorithm that solves element distinctness with $O(N^{2/3})$ queries to $x_1, \ldots, x_N$. This matches the lower bound of [28, 4].

Our algorithm uses a combination of several ideas: quantum search on graphs [2] and quantum walks [24]. While each of those ideas has been used before, the present combination is new.

We first reduce element distinctness to searching a certain graph with vertices $S \subseteq \{1, \ldots, N\}$ as vertices. The goal of the search is to find a marked vertex. Both examining the current vertex and moving to a neighboring vertex cost one time step. (This contrasts with the usual quantum search [21], where only examining the current vertex costs one time step.)

We then search this graph by quantum random walk. We start in a uniform superposition over all vertices of a graph and perform a quantum random walk with one transition rule for unmarked vertices and another transition rule for marked vertices. The result is that the amplitude gathers in the marked vertices and, after $O(N^{2/3})$ steps, the probability of measuring the marked state is a constant.

We also give several extensions of our algorithm. If we have to find whether $x_1, \ldots, x_N$ contain k numbers that are equal: $x_{i_1} = \ldots = x_{i_k}$, we get a quantum algorithm with $O(N^{k/(k+1)})$ queries for any constant[1] k.

If the quantum algorithm is restricted to storing r numbers, $r \leq N^{2/3}$, then we have an algorithm which solves element distinctness with $O(N/\sqrt{r})$ queries which is quadratically better than the classical $O(N^2/r)$ query algorithm. Previously, such quantum algorithm was known only for $r \leq \sqrt{N}$ [10]. For the problem of finding k equal numbers, we get an algorithm that uses $O(\frac{N^{k/2}}{r^{(k-1)/2}})$ queries and stores r numbers, for $r \leq N^{(k-1)/k}$.

For the analysis of our algorithm, we develop a generalization of Grover's algorithm (Lemma 3) which might be of independent interest.

1.1. Related work

Classical element distinctness. Element distinctness has been extensively studied classically. It can be solved with $O(N)$ queries and $O(N \log N)$ time by querying all the elements and sorting them. Then, any two equal elements must be next one to another in the sorted order and can be found by going through the sorted list.

In the usual query model (where one query gives one value of x_i), it is easy to see that $\Omega(N)$ queries are also necessary. Classical lower bounds have also been shown for more general models (e.g. [20]).

If we are restricted to space $S < N$, the running time increases. The straightforward algorithm needs $O(\frac{N^2}{S})$

*School of Mathematics, Institute for Advanced Study, Princeton, NJ 08540, USA, e-mail: ambainis@math.ias.edu. Supported by DARPA and Air Force Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-01-2-0524 (at UC Berkeley) and NSF Grant DMS-0111298 (at IAS).

[1] The big-O constant depends on k. For non-constant k, we can show that the number of queries is $O(k^2 N^{k/(k+1)})$. The proof of that is mostly technical and is omitted in this version.

Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS'04)
0272-5428/04 $20.00 © 2004 **IEEE**

queries. Yao [30] has shown that, for the model of comparison-based branching programs, this is essentially optimal. Namely, any space-S algorithm needs time $T = \Omega(\frac{N^{2-o(1)}}{S})$. For more general models, lower bounds on algorithms with restricted space S is an object of ongoing research [8].

Related problems in quantum computing. In collision problem [12], we are given a 2-1 function f and have to find x, y such that $f(x) = f(y)$. This can be achieved in $O(N^{1/3})$ quantum steps instead of $\Theta(N^{1/2})$ steps classically. $\Omega(N^{1/3})$ is also a quantum lower bound [1, 28, 25].

If element distinctness can be solved with M queries, then collision problem can be solved with $O(\sqrt{M})$ queries [28]. Thus, a quantum algorithm for element distinctness implies a quantum algorithm for collision but not the other way around.

Quantum search on graphs. The idea of quantum search on graphs was proposed by Aaronson and Ambainis [2] for finding a marked item on a d-dimensional grid (problem first considered by Benioff [9]) and other graphs with good expansion properties. Our work has a similar flavor but uses completely different methods to search the graph (quantum walk instead of ''divide-and-conquer").

Quantum walks. There has been considerable amount of research on quantum walks (surveyed in [24]) and their applications (surveyed in [5]). Applications of walks [5] mostly fall into two classes. The first class is exponentially faster hitting times [16, 14, 23]. The second class is quantum walk search algorithms [27, 17, 7].

Our algorithm is most closely related to the second class. In this direction, Shenvi et al. [27] have constructed a counterpart of Grover's search [21] based on quantum walk on hypercube. Childs and Goldstone [17, 18] and Ambainis et al. [7] have used quantum walk on to produce search algorithms on d-dimensional lattices ($d \geq 2$) which is faster than the naive application of Grover's search. This direction is quite closely related to our work. The algorithms by [27, 17, 7] and current paper solve different problems but all have similar structure.

Recent developments. After the work described in this paper, the results and ideas from this paper have been used to construct several other quantum algorithms. Magniez et al. [26] have used our element distinctness algorithm to give an $O(n^{1.3})$ query quantum algorithm for finding triangles in a graph. Ambainis et al. [7] have used ideas from the current paper to construct a faster algorithm for search on 2-dimensional grid. Childs and Eisenberg [15] have given a different analysis of our algorithm.

Szegedy [29] has generalized our results on quantum walk for element distinctness to an arbitrary graph with a large eigenvalue gap and cast them into the language of Markov chains. An advantage of his approach is that it can simultaneously handle any number of solutions (unlike in the present paper which has separate algorithms for single solution case (algorithm 2) and multiple-solution case (algorithm 3)). Furthermore, Szegedy [29] have shown that, for a class of Markov chains, quantum walk algorithms are quadratically faster than the corresponding classical algorithm.

2. Preliminaries

Let $[N]$ denote $\{1, \ldots, N\}$. We consider

Element Distinctness. Given numbers $x_1, \ldots, x_N \in [M]$, are there $i, j \in [N]$, $i \neq j$ such that $x_i = x_j$?

Element k-distinctness. Given numbers $x_1, \ldots, x_N \in [M]$, are there k distinct indices $i_1, \ldots, i_k \in [N]$ such that $x_{i_1} = x_{i_2} = \ldots = x_{i_k}$?

Element distinctness is a particular case of k-distinctness for $k = 2$.

Our model is the quantum query model (for surveys on query model, see [6, 13]). In this model, our goal is to compute a function $f(x_1, \ldots, x_N)$. For example, k-distinctness is viewed as the function $f(x_1, \ldots, x_N)$ which is 1 if there are k distinct $i_1, \ldots, i_k \in [N]$ such that $x_{i_1} = x_{i_2} = \ldots = x_{i_k}$ and 0 otherwise.

The input variables x_i can be accessed by queries to an oracle X and the complexity of f is the number of queries needed to compute f. A quantum computation with T queries is just a sequence of unitary transformations

$$U_0 \to O \to U_1 \to O \to \ldots \to U_{T-1} \to O \to U_T.$$

U_j's can be arbitrary unitary transformations that do not depend on the input bits $x_1, \ldots, x_N$. O are query (oracle) transformations. To define O, we represent basis states as $|i, a, z\rangle$ where i consists of $\lceil \log N \rceil$ bits, a consists of $\lceil \log M \rceil$ quantum bits and z consists of all other bits. Then, O maps $|i, a, z\rangle$ to $|i, (a + x_i) \bmod M, z\rangle$.

In our algorithm, we use queries in two situations. The first situation is when $a = |0\rangle$. Then, the state before the query is some superposition $\sum_{i,z} \alpha_{i,z} |i, 0, z\rangle$ and the state after the query is the same superposition with the information about x_i: $\sum_{i,z} \alpha_{i,z} |i, x_i, z\rangle$. The second situation is when the state before the query is $\sum_{i,z} \alpha_{i,z} |i, -x_i \bmod M, z\rangle$ with the information about information about x_i from a previous query. Then, applying the query transformation makes the state $\sum_{i,z} \alpha_{i,z} |i, 0, z\rangle$, erasing the information about x_i. This can be used to erase the information about x_i from $\sum_{i,z} \alpha_{i,z} |i, x_i, z\rangle$. We first perform a unitary that maps $|x_i\rangle \to |-x_i \bmod M\rangle$, obtaining the state $\sum_{i,z} \alpha_{i,z} |i, -x_i \bmod M, z\rangle$ and then apply the query transformation.

The computation starts with a state $|0\rangle$. Then, we apply $U_0, O, \ldots, O, U_T$ and measure the final state. The result of the computation is the rightmost bit of the state obtained by the measurement.

The quantum computation computes f with bounded error if, for every $x = (x_1, \ldots, x_N)$, the probability that the rightmost bit of $U_T O_x U_{T-1} \ldots O_x U_0 |0\rangle$ equals $f(x_1, \ldots, x_N)$ is at least $1 - \epsilon$ for some fixed $\epsilon < 1/2$.

To simplify the exposition, we occasionally describe a quantum computation as a classical algorithm with several quantum subroutines of the form $U_t O_x U_{t-1} \ldots O_x U_0 |0\rangle$. Any such algorithm can be transformed into an equivalent sequence $U_T O_x U_{T-1} \ldots O_x U_0 |0\rangle$ with the number of queries being equal to the number of queries in the classical algorithm plus the sum of numbers of queries in all quantum subroutines.

Comparison oracle. In a different version of query model, we are only allowed comparison queries. In a comparison query, we give two indices i, j to the oracle. The oracle answers whether $x_i < x_j$ or $x_i \geq x_j$. In the quantum model, we can query the comparison oracle with a superposition $\sum_{i,j} a_{i,j} |i\rangle |j\rangle$. In section 5, we show that our algorithms can be adapted to this model with a logarithmic increase in the number of queries.

3. Results and algorithms

Theorem 1 *Element k-distinctness can be solved by a quantum algorithm with $O(N^{k/(k+1)})$ queries. In particular, element distinctness can be solved by a quantum algorithm with $O(N^{2/3})$ queries.*

Theorem 2 *If memory of a quantum algorithm is restricted to $O(r \log M)$ qubits, then element distinctness can be solved with $O(\max(\frac{N}{\sqrt{r}}, r))$ queries and element k-distinctness can be solved with $O(\max(\frac{N^{k/2}}{r^{(k-1)/2}}, r))$ queries.*

Theorem 1 follows from Theorem 2 by setting $r = \lfloor N^{2/3} \rfloor$ for element distinctness and $r = \lfloor N^{k/(k+1)} \rfloor$ for k-distinctness.

3.1. Main ideas

We start with an informal description of main ideas. For simplicity, we restrict to element distinctness and postpone the more general k-distinctness till the end of this subsection.

Let $r = N^{2/3}$. We define a graph G with $\binom{N}{r} + \binom{N}{r+1}$ vertices. The vertices v_S correspond to sets $S \subseteq [N]$ of size r and $r + 1$. Two vertices v_S and v_T are connected by an edge if $T = S \cup \{i\}$ for some $i \in [N]$. A vertex is marked if S contains i, j, $x_i = x_j$.

Element distinctness reduces to finding a marked vertex in this graph. If we find a marked vertex v_S, then we know that $x_i = x_j$ for some $i, j \in S$, i.e. $x_1, \ldots, x_N$ are not all distinct.

The naive way to find a marked vertex would be to use Grover's quantum search algorithm [21, 11]. If ϵ fraction of vertices are marked, then quantum search finds a marked vertex by examining $O(\frac{1}{\sqrt{\epsilon}})$ vertices. Assume that there exist $i, j \in [N]$ such that $i \neq j$, $x_i = x_j$. For a random S, $|S| = N^{2/3}$, the probability of v_S being marked is

$$Pr[i \in S; j \in S] = Pr[i \in S] Pr[j \in S | i \in S] =$$

$$\frac{N^{2/3}}{N} \frac{N^{2/3} - 1}{N - 1} = (1 - o(1)) \frac{1}{N^{2/3}}.$$

Thus, a quantum algorithm can find a marked vertex by examining $O(\frac{1}{\sqrt{\epsilon}}) = O(N^{1/3})$ vertices. However, to find out if a vertex is marked, the algorithm needs to query $N^{2/3}$ items x_i, $i \in S$. This makes the total query complexity $O(N^{1/3} N^{2/3}) = O(N)$, giving no speedup compared to the classical algorithm which queries all items.

We improve on this naive algorithm by re-using the information from previous queries. Assume that we just checked if v_S is marked by querying all x_i, $i \in S$. If the next vertex v_T is such that T contains only m elements $i \notin S$, then we only need to query m elements x_i, $i \in T - S$ instead of $r = N^{2/3}$ elements x_i, $i \in T$.

To formalize this, we use the following model. At each moment, we are at one vertex of G (superposition of vertices in quantum case). In one time step, we can examine if the current vertex v_S is marked and move to an adjacent vertex v_T. If, in this model, we can find a marked vertex in M steps, then we can solve element distinctness in $M + r$ steps. We first use r queries to query all x_i, $i \in S$ for the starting vertex v_S. Each next step involves moving to an adjacent vertex v_T. This may involve querying one x_i for $i \in T - S$. M moves involve M queries. Checking if the current vertex v_S is marked requires no queries because we already know all x_i, $i \in S$. Thus, the total number of queries is at most $M + r$.

In the next sections, we will show how to search this graph by quantum walk in $O(N^{2/3})$ steps for element distinctness and $O(N^{k/(k+1)})$ steps for k-distinctness.

3.2. The algorithm

Let $x_1, \ldots, x_N \in [M]$. Let r be the number of items that we are allowed to store in our memory. We consider two Hilbert spaces $\mathcal{H}$ and $\mathcal{H}'$. $\mathcal{H}$ has dimension $\binom{N}{r} M^r (N - r)$ and the basis states of $\mathcal{H}$ are $|S, x, y\rangle$ with $S \subseteq [N]$, $|S| = r$, $x \in [M]^r$, $y \in [N] - S$. $\mathcal{H}'$ has dimension $\binom{N}{r+1} M^{r+1} (r+1)$. The basis states of $\mathcal{H}'$ are $|S, x, y\rangle$ with $S \subseteq [N]$, $|S| = r + 1$, $x \in [M]^{r+1}$, $y \in S$.

In the states used by our algorithm, x will always be equal to $(x_{i_1}, \ldots, x_{i_k})$ where $i_1, \ldots, i_k$ are elements of S in increasing order.

1. Apply the transformation mapping $|S\rangle|y\rangle$ to

$$|S\rangle\left(\left(-1+\frac{2}{N-r}\right)|y\rangle+\frac{2}{N-r}\sum_{y'\notin S,y'\neq y}|y'\rangle\right).$$

on the y register of the state in $\mathcal{H}$. (This transformation is a variant of "diffusion transformation" in [21].)

2. Map the state from $\mathcal{H}$ to $\mathcal{H}'$ by adding y to S and changing x to a vector of length $k+1$ by introducing 0 in the location corresponding to y:

3. Query for $f(y)$ and insert it into location of x corresponding to y.

4. Apply the transformation $|S\rangle|y\rangle$

$$|S\rangle\left(\left(-1+\frac{2}{r+1}\right)|y\rangle+\frac{2}{r+1}\sum_{y'\in S,y'\neq y}|y'\rangle\right).$$

on the y register.

5. Erase the element of x corresponding to new y by using it as the input to query for $f(y)$.

6. Map the state back to $\mathcal{H}$ by removing the 0 component corresponding to y from x and removing y from S.

Algorithm 1: One step of quantum walk

We start by defining a quantum walk on $\mathcal{H}$ and $\mathcal{H}'$ (algorithm 1). Each step of the quantum walk starts in a superposition of states in $\mathcal{H}$. The first three steps map the state from $\mathcal{H}$ to $\mathcal{H}'$ and the last three steps map it back to $\mathcal{H}$.

If there at most one set of k values $i_1,\ldots,i_k$ such that $x_{i_1}=x_{i_2}=\ldots=x_{i_k}$, we apply Algorithm 2 ($t_1$ and t_2 are $c_1*\sqrt{r}$ and $c_2*(\frac{N}{r})^{k/2}$ for constants c_1 and c_2 which can be calculated from the analysis in section 4). This algorithm alternates quantum walk with a transformation that changes the phase if the current state contains k equal elements. We give a detailed proof of correctness for Algorithm 2 in section 4 and appendix A.

If there can be more than one set of k equal elements, element k-distinctness is solved by algorithm 3 which samples subsets of x_i and runs algorithm 2 on each subset. A fairly straighforward probability theory argument shows that, with a constant probability, at least one of sampled subsets contains unique $i_1,\ldots,i_k$ satisfying $x_{i_1}=\ldots=x_{i_k}$. This reduces the correctness of Algorithm 3 to the correctness of Algorithm 2. Because of space constraints, the details are omitted in this version.

1. Generate the uniform superposition $\frac{1}{\sqrt{\binom{N}{r}(N-r)}}\sum_{|S|=r,y\notin S}|S\rangle|y\rangle$.

2. Query all x_i for $i\in S$. This transforms the state to

$$\frac{1}{\sqrt{\binom{N}{r}(N-r)}}\sum_{|S|=r,y\notin S}|S\rangle|y\rangle\bigotimes_{i\in S}|x_i\rangle.$$

3. $t_1=O((N/r)^{k/2})$ times repeat:

   (a) Apply phase flip (the transformation $|S\rangle|y\rangle|x\rangle\rightarrow-|S\rangle|y\rangle|x\rangle$) for S such that $x_{i_1}=x_{i_2}=\ldots=x_{i_k}$ for k distinct $i_1,\ldots,i_k\in S$.

   (b) Perform $t_2=O(\sqrt{r})$ steps of the quantum walk (algorithm 1).

4. Measure the final state.

Algorithm 2: Single-solution algorithm

1. Let $T=[N]$.

2. While $|T|>r$ repeat:

   (a) Run Algorithm 2 on x_i, $i\in T$. Measure the final state, obtaining a set S. If there are k equal elements x_i, $i\in S$, stop, answer "not k-distinct".

   (b) For each $i\in T$, remove i from T with probability $\frac{1}{2k+1}$.

3. Query all x_i, $i\in T$ classically. If k equal elements are found, answer "k-distinct", otherwise, answer "not k-distinct".

Algorithm 3: Multiple-solution algorithm

4. Analysis of single-collision algorithm

The number of queries for algorithm 2 is r for creating the initial state and $O((N/r)^{k/2}\sqrt{r})=O(\frac{N^{k/2}}{r^{(k-1)/2}})$ for the rest of the algorithm. Thus, the overall number of queries is $O(\max(r,\frac{N^{k/2}}{r^{(k-1)/2}}))$. The correctness of algorithm 2 follows from

Theorem 3 *Let the input x_1, …, x_N be such that $x_{i_1}=\ldots=x_{i_k}$ for exactly one set of k distinct values $i_1,\ldots,i_k$. With a constant probability, measuring the final state of algorithm 2 gives S such that $i_1,\ldots,i_k\in S$.*

Proof: Let $|S,y\rangle$ be a shortcut for the basis state $|S,x,y\rangle$. (Since the $|x\rangle$ register always contains the state $\otimes_{i\in S}|x_i\rangle$, the state of $|x\rangle$ is uniquely determined by S and y.)

We classify the basis states $|S,y\rangle$ ($|S| = r$, $y \notin S$) into $2k+1$ types. A state $|S,y\rangle$ is of type $(j,0)$ if $|S \cap \{i_1,\ldots,i_k\}| = j$ and $y \notin \{i_1,\ldots,i_k\}$ and of type $(j,1)$ if $|S \cap \{i_1,\ldots,i_k\}| = j$ and $y \in \{i_1,\ldots,i_k\}$. For $j \in \{0,\ldots,k-1\}$, there are both type $(j,0)$ and type $(j,1)$ states. For $j = k$, there are only $(k,0)$ type states. ($(k,1)$ type is impossible because, if, $|S \cap \{i_1,\ldots,i_k\}| = k$, then $y \notin S$ implies $y \notin \{i_1,\ldots,i_k\}$.)

Let $|\psi_{j,l}\rangle$ be the uniform superposition of basis states $|S,y\rangle$ of type (j,l). Let $\tilde{H}$ be the $(2k+1)$-dimensional space spanned by states $|\psi_{j,l}\rangle$.

For the space $\mathcal{H}'$, its basis states $|S,y\rangle$ ($|S| = r+1$, $y \in S$) can be similarly classified into $2k+1$ types. We denote those types (j,l) with $j = |S \cap \{i_1,\ldots,i_k\}|$, $l = 1$ if $y \in \{i_1,\ldots,i_k\}$ and $l = 0$ otherwise. (Notice that, since $y \in S$ for the space $\mathcal{H}'$, we have type $(k,1)$ but no type $(0,1)$.) Let $|\varphi_{j,l}\rangle$ be the uniform superposition of basis states $|S,y\rangle$ of type (j,l) for space $\mathcal{H}'$. Let $\tilde{H}'$ be the $(2k+1)$-dimensional space spanned by $|\varphi_{j,l}\rangle$. Notice that the transformation $|S,y\rangle \to |S \cup \{y\},y\rangle$ maps

$$|\psi_{i,0}\rangle \to |\varphi_{i,0}\rangle,\ |\psi_{i,1}\rangle \to |\varphi_{i+1,1}\rangle.$$

We claim

Lemma 1 *In algorithm 1, steps 1-3 map $\tilde{\mathcal{H}}$ to $\tilde{\mathcal{H}}'$ and steps 4-6 map $\tilde{\mathcal{H}}'$ to $\tilde{\mathcal{H}}$.*

Proof: Omitted. ∎

Thus, algorithm 1 maps $\tilde{\mathcal{H}}$ to itself. Also, in algorithm 2, step 3b maps $|\psi_{k,0}\rangle \to -|\psi_{k,0}\rangle$ and leaves $|\psi_{j,l}\rangle$ for $j < k$ unchanged (because $|\psi_{j,l}\rangle$, $j < k$ are superpositions of states $|S,y\rangle$ which are unchanged by step 3b and $|\psi_{k,0}\rangle$ is a superposition of states $|S,y\rangle$ which are mapped to -$|S,y\rangle$ by step 3b). Thus, every step of algorithm 2 maps $\tilde{\mathcal{H}}$ to itself and it suffices to analyze algorithms 1 and 2 on subspace $\tilde{\mathcal{H}}$.

In this subspace, we will be interested in two particular states. Let $|\psi_{start}\rangle$ be the uniform superposition of all $|S,y\rangle$, $|S| = r$, $y \notin S$. Let $|\psi_{good}\rangle = |\psi_{k,0}\rangle$ be the uniform superposition of all $|S,y\rangle$ with $i \in S$ and $j \in S$. $|\psi_{start}\rangle$ is the algorithm's starting state. $|\psi_{good}\rangle$ is the state we would like to obtain (because measuring $|\psi_{good}\rangle$ gives a random set S such that $\{i_1,\ldots,i_k\} \subseteq S$).

We start by analyzing a single step of quantum walk.

Lemma 2 *Let S be the unitary transformation induced on $\tilde{\mathcal{H}}$ by one step of the quantum walk (algorithm 1). S has $2k+1$ different eigenvalues in $\tilde{\mathcal{H}}$. One of them is 1, with $|\psi_{start}\rangle$ being the eigenvector. The other eigenvalues are $e^{\pm\theta_1 i}$, …, $e^{\pm\theta_k i}$ with $\theta_j = (2\sqrt{j} + o(1))\frac{1}{\sqrt{r}}$.*

Proof: In appendix A.1. ∎

We set $t_2 = \lceil \frac{\pi}{3\sqrt{k}}\sqrt{r} \rceil$. Since one step of quantum walk fixes $\tilde{\mathcal{H}}$, t_2 steps fix $\tilde{\mathcal{H}}$ as well. Moreover, $|\psi_{start}\rangle$ will still be an eigenvector with eigenvalue 1. The other $2k$ eigenvalues become $e^{\pm i(\frac{2\pi\sqrt{j}}{3\sqrt{k}}+o(1))}$. Thus, every of those eigenvalues is a constant independent of N and r.

Let step U_1 be step 3a of algorithm 2 and $U_2 = S^{t_2}$ be step 3b. Then, the entire algorithm consists of applying $(U_2U_1)^{t_1}$ to $|\psi_{start}\rangle$. We will apply

Lemma 3 *Let $\mathcal{H}$ be a finite dimensional Hilbert space and $|\psi_1\rangle$, …, $|\psi_m\rangle$ be an orthonormal basis for $\mathcal{H}$. Let $|\psi_{good}\rangle$, $|\psi_{start}\rangle$ be two states in $\mathcal{H}$ which are superpositions of $|\psi_1\rangle$, …, $|\psi_m\rangle$ with real amplitudes and $\langle\psi_{good}|\psi_{start}\rangle = \alpha$. Let U_1, U_2 be unitary transformations on $\mathcal{H}$ with the following properties:*

1. *U_1 is the transformation that flips the phase on $|\psi_{good}\rangle$ ($U_1|\psi_{good}\rangle = -|\psi_{good}\rangle$) and leaves any state orthogonal to $|\psi_{good}\rangle$ unchanged.*
2. *U_2 is a transformation which is described by a real-valued $m*m$ matrix in the basis $|\psi_1\rangle$, …, $|\psi_m\rangle$. Moreover, $U_2|\psi_{start}\rangle = |\psi_{start}\rangle$ and, if $|\psi\rangle$ is an eigenvector of U_2 perpendicular to $|\psi_{start}\rangle$, then $U_2|\psi\rangle = e^{i\theta}|\psi\rangle$ for $\theta \in [\epsilon, 2\pi-\epsilon]$.*

Then, there exists $t = O(\frac{1}{\alpha})$ such that $\langle\psi_{good}|(U_2U_1)^t|\psi_{start}\rangle = \Omega(1)$. (The constant under $\Omega(1)$ is independent of α but can depend on ϵ.)

Proof: In appendix A.2. ∎

By Lemma 3, we can set $t_1 = O(\frac{1}{\alpha})$ so that the inner product of $(U_2U_1)^{t_1}|\psi_{start}\rangle$ and $|\psi_{good}\rangle$ is a constant. Since $|\psi_{good}\rangle$ is a superposition of $|S,y\rangle$ over S satisfying $\{i_1,\ldots,i_k\} \subseteq S$, measuring $(U_2U_1)^{t_1}|\psi_{start}\rangle$ gives a set S satisfying $\{i_1,\ldots,i_k\} \subseteq S$ with a constant probability.

It remains to calculate α. Let α' be the fraction of S satisfying $\{i_1,\ldots,i_k\} \subseteq S$. Since $|\psi_{start}\rangle$ is the uniform superposition of all $|S,y\rangle$ and $|\psi_{good}\rangle$ is the uniform superposition of $|S,y\rangle$ with $\{i_1,\ldots,i_k\} \subseteq S$ we have $\alpha = \sqrt{\alpha'}$.

$$\alpha' = Pr[\{i_1,\ldots,i_k\} \subseteq S] =$$

$$\frac{r}{N}\prod_{j=1}^{k-1}\frac{r-j}{N-j} = (1-o(1))\frac{r^k}{N^k}.$$

Therefore, $\alpha = \Omega(\frac{r^{k/2}}{N^{k/2}})$ and $t_1 = O((N/r)^{k/2})$. ∎

Lemma 3 might also be interesting by itself. It generalizes one of analyses of Grover's algorithm [3]. Informally, the lemma says that, in Grover-like sequence of transformations $(U_2U_1)^t$, we can significantly relax the constraints on U_2 and the algorithm will still give similar result. It is quite likely that such situations might appear in analysis of other algorithms.

For the quantum walk resulting from element distinctness, Childs and Eisenberg [15] have shown that $\langle\psi_{good}|(U_2U_1)^t|\psi_{start}\rangle$ (and algorithm's success probability) is $1-o(1)$ instead of $\Omega(1)$, given by lemma 3.

5. Running time and other issues

Running time. If non-query transformations are taken into account, the running time of our algorithm is $O(N^{2/3}\log^c N)$ steps, in a sufficiently powerful version of quantum model. The implementation uses a variant of a hash table and is discussed in the full version.

Comparison model. Our algorithm can be adapted to the model of comparison queries similarly to the algorithm of [10]. Instead of having the register $\otimes_{j\in S}|x_j\rangle$, we have a register $|j_1, j_2, \ldots, j_r\rangle$ where $|j_l\rangle$ is the index of the l^{th} smallest element in the set S. Given such register and $y \in [N]$, we can add y to $|j_1, \ldots, j_r\rangle$ by binary search which takes $O(\log\sqrt{N^{2/3}}) = O(\log N)$ queries. We can also remove a given $x \in [N]$ in $O(\log N)$ queries by reversing this process. This gives an algorithm with $O(N^{2/3}\log N)$ queries. It is open whether we can have $O(N^{2/3}\log^c N)$ running time in comparison model.

6. Open problems

1. **Time-space tradeoffs.** Our optimal $O(N^{2/3})$-query algorithm requires space to store $O(N^{2/3})$ items.

   How many queries do we need if algorithm's memory is restricted to r items? Our algorithm needs $O(\frac{N}{\sqrt{r}})$ queries and this is the best known. Curiously, the lower bound for deterministic algorithms in comparison query model is $\Omega(\frac{N^2}{r})$ queries [30] which is quadratically more. This suggests that our algorithm might be optimal in this setting as well. However, the only lower bound is the $\Omega(N^{2/3})$ lower bound for algorithms with unrestricted memory [28].

2. **Optimality of k-distinctness algorithm.** While element distinctness is known to require $\Omega(N^{2/3})$ queries, it is open whether our $O(N^{k/(k+1)})$ query algorithm for k-distinctness is optimal.

   The best lower bound for k-distinctness is $\Omega(N^{2/3})$, by a following argument. We take an instance of element distinctness $x_1, \ldots, x_N$ and transform it into k-distinctness by repeating every element $k-1$ times. If $x_1, \ldots, x_N$ are all distinct, there is no k equal elements. If there are i, j such that $x_i = x_j$ among original N elements, then repeating each of them $k-1$ times creates $2k-2$ equal elements. Therefore, solving k-distinctness on $(k-1)*N$ elements requires at least the same number of queries as solving distinctness on N elements (which requires $\Omega(N^{2/3})$ queries).

3. **Quantum walks on other graphs.** A quantum walk search algorithm based on similar ideas can be used for Grover search on grids [7, 17]. What other graphs can quantum-walks based algorithms search? Is there a graph-theoretic property that determines if quantum walk algorithms work well on this graph?

   [7] and [29] have shown that, for a class of graphs, the performance of quantum walk depends on certain expressions consisting of graph's eigenvalues. In particular, if a graph has a large eigenvalue gap, quantum walk search performs well [29]. A large eigenvalue graph is, however, not necessary, as shown by quantum search algorithms for grids [7, 29].

Acknowledgments. Thanks to Scott Aaronson for showing that k-distinctness is at least as hard as distinctness (remark 2 in section 6) and to Robert Beals, Andrew Childs, Daniel Gottesman, Julia Kempe, Greg Kuperberg, Samuel Kutin, Frederic Magniez, Oded Regev and Mario Szegedy for comments and discussions.

References

[1] S. Aaronson. Quantum lower bound for the collision problem. *Proceedings of STOC'02*, pp. 635-642.

[2] S. Aaronson, A. Ambainis. Quantum search of spatial structures. *Proceedings of FOCS'03*, pp. 200-209.

[3] D. Aharonov. Quantum computation - a review. *Annual Review of Computational Physics(ed. Dietrich Stauffer)*, vol. VI, World Scientific, 1998.

[4] A. Ambainis. Quantum lower bounds for collision and element distinctness with small range, quant-ph/0305179.

[5] A. Ambainis. Quantum walks and their algorithmic applications. *International Journal of Quantum Information*, 1:507-518 (2003).

[6] A. Ambainis. Quantum query algorithms and lower bounds. *Proceedings of FOTFS'III*, to appear. Journal version under preparation.

[7] A. Ambainis, J. Kempe, A. Rivosh. Coins make quantum walks faster, quant-ph/0402107.

[8] P. Beame, M. Saks, X. Sun, E. Vee. Time-space tradeoff lower bounds for randomized computation of decision problems. *Journal of the ACM*, 50: 154-195, 2003. Also FOCS'00.

[9] P. Benioff. Space searches with a quantum robot.*Quantum Computation and Quantum Information: A Millennium Volume*, AMS Contemporary Mathematics Series, 2002. Also quant-ph/0003006.

[10] H. Buhrman, C. Durr, M. Heiligman, P. Hoyer, F. Magniez, M. Santha, and R. de Wolf. Quantum algorithms for element distinctness. *Proceedings of Complexity'01*, pp.131-137,

[11] G. Brassard, P. Høyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation, *Quantum Computation and Quantum Information: A Millennium Volume*, AMS Contemporary Mathematics Series, 2002.

[12] G. Brassard, P. Høyer, and A. Tapp. Quantum cryptanalysis of hash and claw-free functions. Proceedings of LATIN'98, *Lecture Notes in Computer Science*, 1380:163-169, 1998.

[13] H. Buhrman, R. de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288:21-43, 2002.

[14] A. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, D. Spielman Exponential algorithmic speedup by quantum walk. *Proceedings of STOC'03*, pp. 59-68.

[15] A. Childs, J. Eisenberg. Quantum algorithms for subset finding. quant-ph/0311038.

[16] A. Childs, E. Farhi, S. Gutmann. An example of the difference between quantum and classical random walks. *Journal of Quantum Information Processing*, 1:35, 2002.

[17] A. Childs, J. Goldstone. Spatial search by quantum walk, quant-ph/0306054.

[18] A. Childs, J. Goldstone. Spatial search and the Dirac equation, quant-ph/0405120.

[19] E. Farhi, S. Gutmann. Quantum computation and decision trees. *Physical Review A*, 58:915-928, 1998.

[20] D. Grigoriev, M. Karpinski, F. Meyer auf der Heide, and R. Smolensky. A lower bound for randomized algebraic decision trees. *Proceedings of STOC'96*, pp. 612–619.

[21] L. Grover. A fast quantum mechanical algorithm for database search. *Proceedings of STOC'96*, pp. 212-219.

[22] R. Horn, C. Johnson *Matrix Analysis*, Cambridge University Press, 1985.

[23] J. Kempe. Quantum random walks hit exponentially faster Proceedings of RANDOM'03, *Lecture Notes in Computer Science*, 2764:354-369, 2003.

[24] J. Kempe. Quantum random walks - an introductory overview, *Contemporary Physics*, 44:307-327, 2003.

[25] S. Kutin. A quantum lower bound for the collision problem. quant-ph/0304162.

[26] F. Magniez, M. Santha, M. Szegedy. An $O(\tilde{n^{1.3}})$ quantum algorithm for the triangle problem. quant-ph/0310134.

[27] N. Shenvi, J. Kempe, K. Whaley. Quantum random-walk search algorithm, *Physical Review A*, 67:052307, 2003.

[28] Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. *Proceedings of FOCS'02*, pp. 513-519.

[29] M. Szegedy. Spectra of Quantized Walks and a $\sqrt{\delta\epsilon}$ rule, quant-ph/0401053.

[30] A. Yao. Near-optimal time-space tradeoff for element distinctness. *Proceedings of FOCS'88*, pp. 91-97.

A. Proofs of Lemmas for Theorem 3

A.1. Proof of Lemma 2

Proof: [of Lemma 2] We fix a basis for $\tilde{\mathcal{H}}$ consisting of $|\psi_{j,0}\rangle$, $|\psi_{j,1}\rangle$, $j \in \{0,\ldots,j-1\}$ and $|\psi_{k,0}\rangle$ and a basis for $\tilde{\mathcal{H}}'$ consisting of $|\varphi_{0,0}\rangle$ and $|\varphi_{j,1}\rangle$, $|\varphi_{j,0}\rangle$, $j \in \{1,\ldots,m\}$. Let D_ϵ be the matrix

$$D_\epsilon = \begin{pmatrix} 1-2\epsilon & 2\sqrt{\epsilon-\epsilon^2} \\ 2\sqrt{\epsilon-\epsilon^2} & -1+2\epsilon \end{pmatrix}.$$

Claim 1 *Let U_1 be the unitary transformation mapping $\tilde{\mathcal{H}}$ to $\tilde{\mathcal{H}}'$ induced by steps 1-3 of quantum walk. Then, U_1 is described by a block diagonal matrix*

$$U_1 = \begin{pmatrix} D_{\frac{k}{N-r}} & \ldots & 0 & 0 & \\ 0 & D_{\frac{k-1}{N-r}} & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & D_{\frac{1}{N-r}} & 0 \\ 0 & 0 & \ldots & 0 & 1 \end{pmatrix}.$$

Proof: Let $\mathcal{H}_j$ be the 2-dimensional subspace of $\tilde{\mathcal{H}}$ spanned by $|\psi_{j,0}\rangle$ and $|\psi_{j,1}\rangle$. Let $\mathcal{H}'_j$ be the 2-dimensional subspace of $\tilde{\mathcal{H}}'$ spanned by $|\varphi_{j,0}\rangle$ and $|\varphi_{j+1,1}\rangle$.

From the proof of Lemma 1, we know that the subspace $\mathcal{H}_j$ is mapped to the subspace $\mathcal{H}'_j$. Thus, we have a block diagonal matrices with 2*2 blocks mapping $\mathcal{H}_j$ to $\mathcal{H}'_j$ and 1*1 identity matrix mapping $|\psi_{k,0}\rangle$ to $|\varphi_{k,0}\rangle$. It remains to show that the transformation from $\mathcal{H}_j$ to $\mathcal{H}'_j$ is $D_{\frac{k-j}{N-r}}$. Let S

be such that $|S \cap \{i_1, \ldots, i_k\}| = j$. We divide $y \in [N] - S$ into two sets S_0 and S_1.

$$S_0 = \{y : y \in [N] - S, y \notin \{i_1, \ldots, i_k\}\},$$

$$S_1 = \{y : y \in [N] - S, y \in \{i_1, \ldots, i_k\}\}.$$

Since $|S \cap \{i_1, \ldots, i_k\}| = j$, S_1 contains $s_1 = k - j$ elements. Since $S_0 \cup S_1 = [N] - S$ contains $N - r$ elements, S_0 contains $s_0 = N - r - k + j$ elements. Define $|\psi_{S,0}\rangle = \frac{1}{\sqrt{k-j}} \sum_{y \in S_0} |S, y\rangle$ and $|\psi_{S,1}\rangle = \frac{1}{\sqrt{N-r-k+j}} \sum_{y \in S_1} |S, y\rangle$. Then, we have

$$|\psi_{j,0}\rangle = \frac{1}{\sqrt{\binom{r}{j}\binom{N-r}{k-j}}} \sum_{\substack{S:|S|=r, \\ |S \cap \{i_1,\ldots,i_k\}|=j}} |\psi_{S,0}\rangle \qquad (1)$$

and, similarly for $|\psi_{j,1}\rangle$ and $|\psi_{S,1}\rangle$. Fix a particular S for which $|\{i_1, \ldots, i_k\} \cap S| = j$. Then, step 1 of algorithm 1 maps $|\psi_{S,0}\rangle$ to

$$\frac{1}{\sqrt{m_1}} \sum_{y \notin S} \left(\left(-1 + \frac{2}{N-r} \right) |y\rangle + \sum_{y' \neq y, y' \notin S} \frac{2}{N-r} |y'\rangle \right)$$

$$= \frac{1}{\sqrt{m_1}} \left(-1 + \frac{2}{N-r} + (m_1 - 1)\frac{2}{N-r} \right) \sum_{y \in S_0} |y\rangle$$

$$+ (m_1) \frac{1}{\sqrt{m_1}} \frac{2}{N-r} \sum_{y \in S_1} |y\rangle$$

$$= \left(-1 + \frac{2m_1}{N-r} \right) |\psi_{S,0}\rangle + \frac{2\sqrt{m_1}}{N-r} |\psi_{S,1}\rangle$$

$$= \left(1 - \frac{2m_0}{N-r} \right) |\psi_{S,0}\rangle + \frac{2\sqrt{m_1}}{N-r} |\psi_{S,1}\rangle.$$

By a similar calculation (without the last step), $|\psi_{S,1}\rangle$ is mapped to

$$\left(-1 + \frac{2m_0}{N-r} \right) |\psi_{S,1}\rangle + \frac{2\sqrt{m_1}}{N-r} |\psi_{S,0}\rangle.$$

Thus, step 1 produces the transformation $D_{\frac{k-j}{N-r}}$ on $|\psi_{S,0}\rangle$ and $|\psi_{S,1}\rangle$. Since $|\psi_{j,0}\rangle$ and $|\psi_{j,1}\rangle$ are uniform superpositions of $|\psi_{S,0}\rangle$ and $|\psi_{S,1}\rangle$ over all S, step 1 also produces the same transformation $D_{\frac{k-j}{N-r}}$ on $|\psi_{j,0}\rangle$ and $|\psi_{j,1}\rangle$. Steps 2 and 3 just map $|\psi_{j,0}\rangle$ to $|\varphi_{j,0}\rangle$ and $|\psi_{j,1}\rangle$ to $|\varphi_{j+1,1}\rangle$. ∎

Similarly, steps 4-6 give the transformation U_2 described by block-diagonal matrix

$$U_2 = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & D'_{\frac{1}{r+1}} & 0 & \ldots & 0 \\ 0 & 0 & D'_{\frac{2}{r+1}} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & D'_{\frac{k}{r+1}} \end{pmatrix}.$$

from $\tilde{\mathcal{H}}'$ to $\tilde{\mathcal{H}}$. Here, D'_ϵ denotes the matrix

$$D'_\epsilon = \begin{pmatrix} -1 + 2\epsilon & 2\sqrt{\epsilon - \epsilon^2} \\ 2\sqrt{\epsilon - \epsilon^2} & 1 - 2\epsilon \end{pmatrix}.$$

A step of quantum walk is $S = U_2 U_1$. Let V be the diagonal matrix with odd entries on the diagonal being 1 and even entries being -1. Since $V^2 = I$, we have $S = U_2 V^2 U_1 = U'_2 U'_1$ for $U'_2 = U_2 V$ and $U'_1 = V U_1$. Let

$$E_\epsilon = \begin{pmatrix} 1 - 2\epsilon & -2\sqrt{\epsilon - \epsilon^2} \\ 2\sqrt{\epsilon - \epsilon^2} & 1 - 2\epsilon \end{pmatrix},$$

$$E'_\epsilon = \begin{pmatrix} 1 - 2\epsilon & 2\sqrt{\epsilon - \epsilon^2} \\ -2\sqrt{\epsilon - \epsilon^2} & 1 - 2\epsilon \end{pmatrix},$$

Then, U'_1 and U'_2 are equal to U_1 and U_2, with every D_ϵ or D'_ϵ replaced by corresponding E_ϵ or E'_ϵ. We will first diagonalize U'_1 and U'_2 separately and then argue that eigenvalues of $U'_2 U'_1$ are almost the same as eigenvalues of U'_2.

Since U'_2 is block diagonal, it suffices to diagonalize each block. 1*1 identity block has eigenvalue 1. For a matrix E_ϵ, its characteristic polynomial is $\lambda^2 - (2 - 4\epsilon)\lambda + 1 = 0$ and its roots are $2\epsilon \pm \sqrt{1 - 4\epsilon^2} i$. For $\epsilon = o(1)$, this is equal to $e^{\pm(2+o(1))i\sqrt{\epsilon}}$. Thus, the eigenvalues of U'_2 are 1, and $e^{\pm 2\frac{\sqrt{j}}{\sqrt{r+1}} i}$ for $j \in \{1, 2, \ldots, k\}$. Similarly, the eigenvalues of U'_1 are 1, and $e^{\pm 2\frac{\sqrt{j}}{\sqrt{N-r}} i}$ for $j \in \{1, 2, \ldots, k\}$.

To complete the proof, we use the following bound on the eigenvalues of the product of two matrices. (It can be derived from Hoffman-Wielandt theorem in matrix analysis [22]. We omit the derivation due to space constraints.)

Theorem 4 *Let A and B be unitary matrices. Assume that A has eigenvalues $1 + \delta_1$, …, $1 + \delta_m$, B has eigenvalues μ_1, …, μ_m and AB has eigenvalues μ'_1, …, μ'_m. Then,*

$$|\mu_i - \mu'_i| \leq \sum_{i=1}^{m} |\delta_i|.$$

Let $A = U'_1$ and $B = U'_2$. Since $|e^{\epsilon i} - 1| \leq |\epsilon|$, each of $|\delta_i|$ is of order $O(\frac{1}{\sqrt{N-r}})$. Therefore, their sum is of order $O(\frac{1}{\sqrt{N-r}})$ as well. Thus, for each eigenvalue of U'_2, there is a corresponding eigenvalue of $U'_1 U'_2$ that differs by at most by $O(\frac{1}{\sqrt{N-r}})$. The lemma now follows from $\frac{1}{\sqrt{N-r}} = o(\frac{1}{\sqrt{r+1}})$. ∎

A.2. Proof of Lemma 3

We assume that α is less than some small constant (for example, $\alpha < 0.1$). Otherwise, we can just take $t = 0$ and get $\langle \psi_{good} | (U_2 U_1)^t | \psi_{start} \rangle = \langle \psi_{good} | \psi_{start} \rangle \geq 0.1$.

Consider the eigenvalues of U_2. Since U_2 is described by a real $m * m$ matrix (in the basis $|\psi_1\rangle$, …, $|\psi_m\rangle$), its

characteristic polynomial has real coefficients. Therefore, the eigenvalues are 1, -1, $e^{\pm i\theta_1}$, ..., $e^{\pm i\theta_l}$. For simplicity, assume that there is no -1 eigenvalue.

Let $|w_{j,+}\rangle$, $|w_{j,-}\rangle$ be the eigenvectors of U_2 with eigenvalues $e^{i\theta_j}$, $e^{-i\theta_j}$. Let $|w_{j,+}\rangle = \sum_{j'=1}^{l} c_{j,j'}|\psi_{j'}\rangle$. Then, we can assume that $|w_{j,-}\rangle = \sum_{j'=1}^{l} c^*_{j,j'}|\psi_{j'}\rangle$. (Since U_2 is a real matrix, taking $U_2|w_{j,+}\rangle = e^{i\theta_j}|w_{j,+}\rangle$ and replacing every number with its complex conjugate gives $U_2|w\rangle = e^{-i\theta_j}|w\rangle$ for $|w\rangle = \sum_{j=1}^{l} c^*_{j,j'}|\psi_{j'}\rangle$.)

We write $|\psi_{good}\rangle$ in a basis consisting of eigenvectors of U_2:

$$|\psi_{good}\rangle = \alpha|\psi_{start}\rangle + \sum_{j=1}^{l}(a_{j,+}|w_{j,+}\rangle + a_{j,-}|w_{j,-}\rangle). \quad (2)$$

W. l. o. g., assume that α and all $a_{j,+}$, $a_{j,-}$ are positive reals. (Otherwise, multiply $|w_{j,+}\rangle$ and $|w_{j,-}\rangle$ by appropriate complex factors to make $a_{j,+}$ or $a_{j,-}$ real.)

We also have $a_{j,+} = a_{j,-}$. (To see that, let $|\psi_{good}\rangle = \sum_{j'=1}^{l} b_{j'}|\psi_{j'}\rangle$. Then, $b_{j'}$ are real (because vector $|\psi_{good}\rangle$ and matrix U are both real). We have $\langle\psi_{good}|w_{j,+}\rangle = a_{j,+} = \sum_{j'=1}^{l} b_{j'}c^*_{j,j'}$ and $\langle\psi_{good}|w_{j,-}\rangle = a_{j,-} = \sum_{j'=1}^{l} b_{j'}(c^*_{j,j'})^* = (\sum_{j'=1}^{l} b_{j'}c^*_{j,j'})^* = a^*_{j,+}$. Since $a_{j,+}$ is real, this means $a_{j,+} = a_{j,-}$.) Therefore, we denote $a_{j,+} = a_{j,-}$ as simply a_j.

Consider the vector

$$|v_\beta\rangle = \alpha\left(1 + i\cot\frac{\beta}{2}\right)|\psi_{start}\rangle +$$
$$\sum_{j=1}^{l} a_j\left(1 + i\cot\frac{-\theta_j+\beta}{2}\right)|w_{j,+}\rangle +$$
$$\sum_{j=1}^{l} a_j\left(1 + i\cot\frac{\theta_j+\beta}{2}\right)|w_{j,-}\rangle. \quad (3)$$

We have $|v_\beta\rangle = |\psi_{good}\rangle + i|v'_\beta\rangle$ where

$$|v'_\beta\rangle = \alpha\cot\frac{\beta}{2}|\psi_{start}\rangle + \sum_{j=1}^{l} a_j\cot\frac{-\theta_j+\beta}{2}|w_{j,+}\rangle +$$
$$\sum_{j=1}^{l} a_j\cot\frac{\theta_j+\beta}{2}|w_{j,-}\rangle.$$

Claim 2 *If $|v'_\beta\rangle$ is orthogonal to $|\psi_{good}\rangle$, then $|v_\beta\rangle$ is an eigenvector of U_2U_1 with an eigenvalue of $e^{i\beta}$ and $|v_{-\beta}\rangle$ is an eigenvector of U_2U_1 with an eigenvalue of $e^{-i\beta}$.*

Proof: Since $|v'_\beta\rangle$ is orthogonal to $|\psi_{good}\rangle$, we have $U_1|v'_\beta\rangle = |v'_\beta\rangle$ and $U_1|v_\beta\rangle = -|\psi_{good}\rangle + i|v'_\beta\rangle$. Therefore,

$$U_2U_1|v_\beta\rangle = \alpha\left(-1 + i\cot\frac{\beta}{2}\right)|\psi_{start}\rangle +$$
$$\sum_{j=1}^{l} a_j e^{i\theta_j}\left(-1 + i\cot\frac{-\theta_j+\beta}{2}\right)|w_{j,+}\rangle +$$
$$\sum_{j=1}^{l} a_j e^{-i\theta_j}\left(-1 + i\cot\frac{\theta_j+\beta}{2}\right)|w_{j,-}\rangle.$$

Furthermore,

$$1 + i\cot x = \frac{\sin x + i\cos x}{\sin x} = \frac{e^{i(\frac{\pi}{2}-x)}}{\sin x},$$
$$-1 + i\cot x = \frac{-\sin x + i\cos x}{\sin x} = \frac{e^{i(\frac{\pi}{2}+x)}}{\sin x},$$

Therefore,

$$\left(-1 + i\cot\frac{\beta}{2}\right) = e^{i\beta}\left(1 + i\cot\frac{\beta}{2}\right),$$
$$e^{i\theta_j}\left(-1 + i\cot\frac{-\theta_j+\beta}{2}\right) = \frac{e^{i(\frac{\pi}{2}+\frac{\theta_j}{2}+\frac{\beta}{2})}}{\sin\frac{-\theta_j+\beta}{2}} =$$
$$e^{i\beta}\left(1 + i\cot\frac{-\theta_j+\beta}{2}\right)$$

and similarly for the coefficient of $|w_{j,-}\rangle$. This means that $U_2U_1|v_\beta\rangle = e^{i\beta}|v_\beta\rangle$.

For $|v_{-\beta}\rangle$, we write out the inner products $\langle\psi_{good}|v'_\beta\rangle$ and $\langle\psi_{good}|v'_{-\beta}\rangle$. Then, we see that $\langle\psi_{good}|v'_{-\beta}\rangle = -\langle\psi_{good}|v'_\beta\rangle$. Therefore, if $|\psi_{good}\rangle$ and $|v'_\beta\rangle$ are orthogonal, so are $|\psi_{good}\rangle$ and $|v'_{-\beta}\rangle$. By the argument above, this implies that $|v_{-\beta}\rangle$ is an eigenvector of U_2U_1 with an eigenvalue $e^{-i\beta}$. ∎

Claim 3 *There exists $\beta = \Omega(\alpha)$ such that $|v'_\beta\rangle$ is orthogonal to $|\psi_{good}\rangle$.*

Proof: Let $f(\beta) = \langle\psi_{good}|v'_\beta\rangle$. We have $f(\beta) = f_1(\beta) + f_2(\beta)$ where $f_1(\beta) = \alpha^2\cot\frac{\beta}{2}$ and

$$f_2(\beta) = \sum_{j=1}^{l} a_j^2\left(\cot\frac{-\theta_j+\beta}{2} + \cot\frac{\theta_j+\beta}{2}\right).$$

We estimate each of f_1 and f_2 separately. First, we have

$$\cot\frac{\beta}{2} = \frac{1}{\tan\frac{\beta}{2}} = \frac{1}{\frac{\beta}{2} + O(\beta^2)} = \frac{2}{\beta} + O(1).$$

Second,

$$\cot\frac{-\theta_j+\beta}{2} + \cot\frac{\theta_j+\beta}{2} =$$
$$-\frac{\sin\beta}{\sin\frac{\theta_j+\beta}{2}\sin\frac{\theta_j-\beta}{2}} = -\frac{\beta}{\sin^2\frac{\theta_j}{2}} + O(\beta^2).$$

Therefore, $f_2(\beta) = -A\beta + O(\beta^2)$ where $A = \sum_{j=1}^{l}\frac{a_j^2}{\sin^2\frac{\theta_j}{2}}$. We have $A = O(1)$. (Since $\theta_j \in [\epsilon, \pi-\epsilon]$,

$A \leq \frac{1}{\sin^2\epsilon}\sum_{j=1}^{l} a_j^2$. Since $\|s\|^2 = \alpha^2 + 2\sum_{j=1}^{l} a_j^2$, we have $\sum_{j=1}^{l} a_j^2 \leq \frac{1}{2}$ and $A \leq \frac{1}{2\sin^2\epsilon}$.)

Putting all together, we have that $f(\beta)$ is equal to $\frac{2\alpha^2}{\beta} - A\beta$ plus lower order terms. Therefore, $f(\beta) = 0$ for $\beta = (1+o(1))\frac{\sqrt{2}\alpha}{\sqrt{A}} = \Theta(\alpha)$. ∎

Let $|u_1\rangle = \frac{|v_\beta\rangle}{\|v_\beta\|}$ and $|u_2\rangle = \frac{|v_{-\beta}\rangle}{\|v_{-\beta}\|}$. We will show that $|\psi_{start}\rangle$ is almost a linear combination of $|u_1\rangle$ and $|u_2\rangle$. Define $|\psi_{end}\rangle = \frac{|v_{end}\rangle}{\|v_{end}\|}$ where

$$|v_{end}\rangle = \sum_{j=1}^{l} a_j \left(1 + i\cot\frac{-\theta_j}{2}\right)|w_{j,+}\rangle + \sum_{j=1}^{l} a_j \left(1 + i\cot\frac{\theta_j}{2}\right)|w_{j,-}\rangle. \quad (4)$$

Claim 4

$$|u_1\rangle = c_{start} i|\psi_{start}\rangle + c_{end}|\psi_{end}\rangle + |u_1'\rangle,$$

$$|u_2\rangle = -c_{start} i|\psi_{start}\rangle + c_{end}|\psi_{end}\rangle + |u_2'\rangle$$

for some c_{start}, c_{end}, $\|u_1'\| = O(\beta)$ *and* $\|u_2'\| = O(\beta)$.

Proof: By regrouping terms in equation (3), we have

$$|v_\beta\rangle = \alpha i\cot\frac{\beta}{2}|\psi_{start}\rangle + |v_{end}\rangle + |v_\beta''\rangle \quad (5)$$

where

$$|v_\beta''\rangle = \alpha|\psi_{start}\rangle + \sum_{j=1}^{l} a_j i\left(\cot\frac{-\theta_j+\beta}{2} - \cot\frac{-\theta_j}{2}\right)|w_{j,+}\rangle + \sum_{j=1}^{l} a_j i\left(\cot\frac{\theta_j+\beta}{2} - \cot\frac{\theta_j}{2}\right)|w_{j,-}\rangle.$$

We claim that $\|v_\beta''\| = O(\beta)\|v_\beta\|$. We prove this by showing that the absolute value of each of coefficients in $|v_\beta''\rangle$ is $O(\beta)$ times the absolute value of corresponding coefficient in $|v_\beta\rangle$. The coefficient of $|\psi_{start}\rangle$ is α in $|v_\beta'\rangle$ and $\alpha(1+i\cot\frac{\beta}{2})$ in $|v_\beta\rangle$. For small β, $\cot\frac{\beta}{2}$ is of order $\Omega(\frac{1}{\beta})$. Therefore, $\alpha = O(\beta)|\alpha(1+i\cot\frac{\beta}{2})|$. For the coefficient of $|w_{j,+}\rangle$, we have

$$\cot\frac{-\theta_j+\beta}{2} - \cot\frac{-\theta_j}{2} = \frac{\sin\frac{\beta}{2}}{\sin\frac{-\theta_j+\beta}{2}\sin\frac{-\theta_j}{2}}.$$

For small β, this is approximately $\frac{\beta}{2\sin^2\frac{-\theta_j}{2}}$. Since $\theta_j \in [\epsilon, \pi-\epsilon]$, we have $|\sin\frac{-\theta_j}{2}| \geq \sin\frac{\epsilon}{2}$ and $\frac{\beta}{2\sin^2\frac{-\theta_j}{2}} \leq \frac{\beta}{2\sin^2\frac{\epsilon}{2}} = O(\beta)$. Therefore, the absolute value of coefficient of $|w_{j,+}\rangle$ in $|v_\beta''\rangle$ is $O(\beta) * a_j$. On the other hand, the coefficient of $|w_{j,+}\rangle$ in $|v_\beta\rangle$ is $a_j(1+i\cot\frac{-\theta_j+\beta}{2})$ which is of absolute value at least a_j. Similarly, we can bound the absolute value of coefficient of $|w_{j,-}\rangle$.

By dividing equation (5) by $\|v_\beta\|$, we get

$$|u_1\rangle = c_{start} i|\psi_{start}\rangle + c_{end}|\psi_{end}\rangle + |u_1'\rangle$$

for $c_{start} = \frac{\alpha\cot\frac{\beta}{2}}{\|v_\beta\|}$, $c_{end} = \frac{\|v_{end}\|}{\|v_\beta\|}$ and $|u_1'\rangle = \frac{1}{\|v_\beta\|}|v_\beta''\rangle$. Since $\|v_\beta''\| = O(\beta)\|v_\beta\|$, we have $\|u_1'\| = O(\beta)$. The proof for u_2 is similar. ∎

Since $|u_1\rangle$ and $|u_2\rangle$ are eigenvectors of U_2U_1 with different eigenvalues, they must be orthogonal. Therefore,

$$\langle u_1|u_2\rangle = -c_{start}^2 + c_{end}^2 + O(\beta) = 0.$$

Also,

$$\|u_1\|^2 = c_{start}^2 + c_{end}^2 + O(\beta^2) = 1.$$

These two equalities together imply that $c_{start} = \frac{1}{\sqrt{2}} + O(\beta)$ and $c_{end} = \frac{1}{\sqrt{2}} + O(\beta)$. Therefore,

$$|u_1\rangle = \frac{1}{\sqrt{2}} i|\psi_{start}\rangle + \frac{1}{\sqrt{2}}|\psi_{end}\rangle + |u_1''\rangle,$$

$$|u_2\rangle = -\frac{1}{\sqrt{2}} i|\psi_{start}\rangle + \frac{1}{\sqrt{2}}|\psi_{end}\rangle + |u_2''\rangle,$$

with $\|u_1''\| = O(\beta)$ and $\|u_2''\| = O(\beta)$. This means that

$$|\psi_{start}\rangle = -\frac{i}{\sqrt{2}}|u_1\rangle + \frac{i}{\sqrt{2}}|u_2\rangle + |w'\rangle,$$

$\|w'\| = O(\beta)$. Let $t = \lfloor\frac{\pi}{2\beta}\rfloor$. Then, $(U_2U_1)^t|u_1\rangle$ is almost $i|u_1\rangle$ (plus a term of order $O(\beta)$) and $(U_2U_1)^t|u_2\rangle$ is almost $-i|u_2\rangle$. Therefore,

$$(U_2U_1)^t|\psi_{start}\rangle = |\psi_{end}\rangle + |v'\rangle$$

where $\|v'\| = O(\beta)$. This means that the inner product $\langle\psi_{good}|(U_2U_1)^t|\psi_{start}\rangle$ is $\langle\psi_{good}|\psi_{end}\rangle$ plus a term which is $O(\beta)$. To complete the proof, it suffices to show

Claim 5 $\langle\psi_{good}|\psi_{end}\rangle = \Omega(1)$.

Proof: Since $|\psi_{end}\rangle = \frac{|v_{end}\rangle}{\|v_{end}\|}$, we have $\langle\psi_{good}|\psi_{end}\rangle = \frac{\langle\psi_{good}|v_{end}\rangle}{\|v_{end}\|}$. By definition of $|v_{end}\rangle$ (equation (4)), $\langle\psi_{good}|v_{end}\rangle = 2\sum_{j=1}^{l} a_j^2$. By equation (2), $\|\psi_{good}\|^2 = \alpha^2 + 2\sum_{j=1}^{l} a_j^2$. Since $\|\psi_{good}\|^2 = 1$, we have $\langle\psi_{good}|v_{end}\rangle = 1 - \alpha^2$. Since $\alpha < 0.1$ (see beginning of the proof), we have $\langle\psi_{good}|\psi_{end}\rangle \geq \frac{0.99}{\|v_{end}\|}$.

We have $\|v_{end}\|^2 = a_0^2 + 2\sum_{j=1}^{l} a_j^2(1+\cot^2\frac{\theta_j}{2})$. Since $\theta_k \in [\epsilon, \pi-\epsilon]$, $\|v_{end}\|^2 \leq a_0^2 + 2\sum_{j=1}^{l} a_j^2(1+\cot^2\epsilon) \leq (1+\cot^2\epsilon)$ and $\langle\psi_{good}|\psi_{end}\rangle \geq \frac{0.99}{\sqrt{1+\cot^2\epsilon}}$ which is a constant if ϵ is fixed. ∎