Electronic Colloquium on Computational Complexity, Report No. 121 (2010)

Inverting a permutation is as hard as unordered search *

Ashwin Nayak †
U. Waterloo & Perimeter

July 16, 2010

Abstract

We describe a reduction from the problem of unordered search (with a unique solution) to the problem of inverting a permutation. Since there is a straightforward reduction in the reverse direction, the problems are essentially equivalent.

The reduction helps us bypass the Bennett-Bernstein-Brassard-Vazirani hybrid argument [2] and the Ambainis quantum adversary method [1] that were earlier used to derive lower bounds on the quantum query complexity of the problem of inverting permutations. It directly implies that the quantum query complexity of the problem is in $\Omega(\sqrt{n})$.

1 Introduction

Let n be an even positive integer. The problem $\textsc{Permutation}_n$ of inverting a permutation π on the set $[n] = \{1, 2, \ldots, n\}$ is defined as follows. Given π in the form of an oracle, and n as input, output "yes" if the pre-image $\pi^{-1}(1)$ is even and "no" if it is odd. A related problem is that of unordered search: Given a function $f : [n] \to \{0, 1\}$ as an oracle, and n as input, output "yes" if $f^{-1}(1)$ is non-empty and "no" otherwise. In other words, determine if f maps any element $i \in [n]$ to 1. In this article, we restrict ourselves to functions f which map at most one element to 1. As we might expect, these constitute the hardest instances of unordered search. We refer to the corresponding sub-problem as $\textsc{Unique Search}_n$. For further background on these problems and the oracle model of computation, we refer the reader to [2, 1].

An algorithm for unordered search (in fact, for $\textsc{Unique Search}_n$) may be used to solve the inversion problem $\textsc{Permutation}_n$ in the obvious manner, using the same number of oracle queries. Namely, we define a boolean function f on $[n]$ such that $f(i) = 1$ iff $\pi(i) = 1$ and i is even. This function may be evaluated with one classical query to an oracle for π. An additional query is required in the quantum case to "erase" the answer to the first query. Therefore the Grover quantum search algorithm [3] solves this problem with $\mathrm{O}(\sqrt{n})$ queries to an oracle for π. We describe a reduction in the reverse direction, i.e., we show that any algorithm that solves $\textsc{Permutation}_n$ also solves $\textsc{Unique Search}_{n/2}$. More accurately, the reduction is between distributional versions of the problems, with equal weight on on "yes" and "no" instances, and with uniform conditional distributions for each kind of instance. Due to the inherent symmetry in the two problems under consideration, the distributional and worst case versions of the problems are, in fact, equivalent in query complexity, up to constant factors (for the distributions described above). We elaborate on this below.

Let μ be the distribution which assigns probability $1/2$ to the constant function 0, and probability $1/2n$ to each of the "yes" instances of $\textsc{Unique Search}_n$.

*The reduction reported in this note was discovered in 2004 and communicated informally to a few people. It is written up here for wider dissemination.

†Department of Combinatorics and Optimization, and Institute for Quantum Computing, University of Waterloo, 200 University Ave. W., Waterloo, ON N2L 3G1, Canada. E-mail: ashwin.nayak@uwaterloo.ca. Research supported in part by NSERC Canada, CIFAR, an ERA (Ontario), QuantumWorks, MITACS, and ARO (USA). A.N. is also Associate Faculty, Perimeter Institute for Theoretical Physics, Waterloo, Canada. Research at Perimeter Institute for Theoretical Physics is supported in part by the Government of Canada through Industry Canada and by the Province of Ontario through MRI.

ISSN 1433-8092

Theorem 1.1. *Let n be an even positive integer. Let $\mathcal{A}$ be an (classical or quantum) algorithm that solves* Permutation$_n$ *with distributional error at most $\varepsilon < 1/2$ on the uniform distribution over permutations on $[n]$, with Q queries to the permutation oracle. Then there is an algorithm of the same kind as $\mathcal{A}$, that solves* Unique Search$_{n/2}$ *with distributional error at most $\frac{1}{3-2\varepsilon} < 1/2$ with respect to μ, with at most Q queries to the search oracle in the classical case, and at most $2Q$ queries in the quantum case.*

An algorithm that is correct with probability $1-\varepsilon$ in the worst case implies an an algorithm with distributional correctness probability $1-\varepsilon$ with respect to any distribution. Conversely, composing a function $f : [n] \to \{0,1\}$ with a permutation σ on $[n]$ preserves the "yes" and the "no" instances of Unique Search$_n$. Therefore, an algorithm $\mathcal{B}$ with distributional error at most ε with respect to the distribution μ gives us an algorithm with worst case error at most 2ε: We pick a uniformly random permutation σ on $[n]$, and then run the algorithm $\mathcal{B}$ with every oracle query $i \in [n]$ replaced by a query to $\sigma(i)$. Effectively, any single instance of Unique Search$_n$ is mapped to a uniformly random instance with the same answer. Since $\mathcal{B}$ makes an error with probability at most 2ε on uniformly random "yes" or "no" instances alone, the worst case error of the new algorithm is at most 2ε.

Similarly, composing a permutation π on $[n]$ with another permutation that permutes odd integers among themselves and even integers among themselves also preserves the "yes" and the "no" instances of Permutation$_n$. Therefore, an algorithm with distributional error at most ε with respect to the uniform distribution for Permutation$_n$ implies an algorithm with worst case probability of error at most 2ε.

As Unique Search$_{n/2}$ requires $\Omega(\sqrt{n}\,)$ queries [2] (in fact, even in the distributional case), for any constant probability of error $\delta < 1/2$, Theorem 1.1 implies that any quantum algorithm for Permutation$_n$ with oracle access to the permutation requires $\Omega(\sqrt{n})$ queries. The lower bound of $\Omega(n)$ in the classical case is straightforward.

The inversion problem Permutation was originally used by Bennett, Brassard, Bernstein, and Vazirani [2] to show that relative to a random permutation oracle A, with probability 1, $\mathbf{NP}^A \cap \mathbf{co\text{-}NP}^A \not\subseteq \mathbf{BQP}^A$. Using a nested hybrid argument, they showed that the inversion problem requires $\Omega(\sqrt[3]{n}\,)$ queries (for constant probability of error under the uniform distribution). The optimal bound of $\Omega(\sqrt{n}\,)$ was established (for worst case query complexity) by Ambainis [1] using the then newly minted quantum adversary method. The reduction we present bypasses these techniques and shows a direct connection between inversion and search.

2 The reduction

As it makes no reference to the model of computation (be it classical randomized, or quantum), we skip ahead to the presentation of the reduction, thereby proving Theorem 1.1. We present it in a non-constructive "top-down" fashion to elucidate the intuition, and then sketch the reduction.

Given an algorithm for inverting a permutation on $[n]$ (specified by an oracle), we devise an algorithm for unique unordered search on $[n/2]$. Assume that the algorithm $\mathcal{A}$ identifies uniformly random permutations chosen from one of the two following sets:

$$\begin{aligned} P_0 &= \left\{ \pi \ : \ \pi \text{ is a permutation on } [n], \quad \pi^{-1}(1) \text{ is odd } \right\} \\ P_1 &= \left\{ \pi \ : \ \pi \text{ is a permutation on } [n], \quad \pi^{-1}(1) \text{ is even } \right\} \ , \end{aligned}$$

with distributional error ε. We have $|P_0| = |P_1| = n!/2$.

We run the algorithm on oracles not in its domain, a device first employed in [4], and in numerous subsequent works to great effect. These oracles compute functions $h : [n] \to [n]$ with a unique collision at 1, with one odd and one even number in the colliding pair. In other words, the function is such that there are precisely two distinct elements i, j with the same image under h, this image is 1, and precisely one of i, j is odd (and the other is even). Let Q denote the set of all such functions. We have $|Q| = \frac{n}{4} n!$.

Since the algorithm $\mathcal{A}$ distinguishes (on average) a random permutation from P_0 from a random permutation from P_1, the algorithm necessarily also accomplishes at least one of the following tasks (on average).

> **Task 1:**
> Distinguish a uniformly random permutation from P_0 from a uniformly random function from Q.

Task 2:
Distinguish a uniformly random permutation from P_1 from a uniformly random function from Q.

Formally, this is a consequence of the triangle inequality for norms; the ℓ_1 norm in the case of classical algorithms and the trace norm in the case of quantum computation. For sake of concreteness, let the algorithm $\mathcal{A}$ be quantum. Let the final states of the algorithm on a uniformly random input from P_0, P_1 and Q be denoted by ρ_0, ρ_1 and ρ, respectively. From the correctness of the algorithm, we have

$$2(1-2\varepsilon) \quad \leq \quad \|\rho_0 - \rho_1\|_{\mathrm{tr}} \quad .$$

From the triangle inequality,

$$\|\rho_0 - \rho_1\|_{\mathrm{tr}} \quad \leq \quad \|\rho_0 - \rho\|_{\mathrm{tr}} + \|\rho_1 - \rho\|_{\mathrm{tr}} \,,$$

whereby at least one of $\|\rho_a - \rho\|_{\mathrm{tr}}$, $a = 0, 1$, is at least $(1-2\varepsilon) > 0$. Equivalently, the algorithm solves at least one of the two tasks above. We show that in either case, we get an algorithm for unique unordered search.

Consider any fixed permutation π on $[n]$. Consider also the functions in Q that differ from π in exactly one point. These are functions h with a unique collision such that the collision is at 1, and $\left|\pi^{-1}(1) \cap h^{-1}(1)\right| = 1$. If $\pi \in P_0$, then the even element that is also mapped to 1 by h is precisely the one on which π and h differ. Similarly, if $\pi \in P_1$, then the odd element that is also mapped to 1 by h is precisely the one on which π and h differ. Let Q_π denote the set of such functions h. We have $|Q_\pi| = n/2$.

Without loss of generality, assume that the algorithm solves Task 1. If we pick a uniformly random permutation $\pi \in P_0$, and then pick a uniformly random function h in Q_π, then h is uniformly random in Q. In particular,

$$\rho \quad = \quad \frac{1}{|Q|} \sum_{h \in Q} \rho_h \quad = \quad \frac{2}{n!} \sum_{\pi \in P_0} \frac{2}{n} \sum_{h \in Q_\pi} \rho_h \,,$$

where ρ_g denotes the final state of the algorithm on oracle input g.

Since the algorithm $\mathcal{A}$ solves Task 1, it also distinguishes ρ_π from a uniformly random $h \in Q_\pi$ for at least one π. Formally,

$$1 - 2\varepsilon \quad \leq \quad \|\rho_0 - \rho\|_{\mathrm{tr}} \quad \leq \quad \frac{2}{n!} \sum_{\pi \in P_0} \left\| \rho_\pi - \frac{2}{n} \sum_{h \in Q_\pi} \rho_h \right\|_{\mathrm{tr}} \quad ,$$

so we have that for at least one $\pi \in P_0$,

$$1 - 2\varepsilon \quad \leq \quad \left\| \rho_\pi - \frac{2}{n} \sum_{h \in Q_\pi} \rho_h \right\|_{\mathrm{tr}} \quad .$$

Finally, we note that for any fixed permutation $\pi \in P_0$, distinguishing π from a function from Q_π is equivalent to Unique Search$_{n/2}$. Given an oracle f for the latter, defined on $[n/2]$, we define a function $h_{\pi,f} : [n] \to [n]$ as follows. Let $h_{\pi,f}(i) = \pi(i)$ for all odd i, and also for those even i with $f(i/2) = 0$. For the lone even i, if any, with $f(i/2) = 1$, set $h_{\pi,f}(i) = 1$. The function $h_{\pi,f}$ coincides with π if $f^{-1}(1)$ is empty, and belongs to Q_π otherwise. Moreover, $h_{\pi,f}$ may be evaluated with two queries to an oracle for f. So the algorithm $\mathcal{A}$ for inverting a permutation on $[n]$ may be used to solve unique unordered search on $[n/2]$ with probability of correctness at least $\frac{1}{2} + \frac{1}{4}(1-2\varepsilon)$ under the distribution μ.

While the above argument suffices to prove a query lower bound for Permutation$_n$, it only gives us a non-constructive proof of existence of an algorithm for Unique Search$_{n/2}$. We may however easily convert the proof into a reduction in the formal sense. Suppose we are given an algorithm $\mathcal{A}$ as in the statement of Theorem 1.1 that takes as input an even integer $n \geq 1$, and an oracle $g : [n] \to [n]$. Then the reduction presented as Algorithm 1 solves Unique Search$_{n/2}$ with probability $\frac{1}{3-2\varepsilon}$.

We may calculate the probability of error of this algorithm by considering "yes" and "no" instances separately. The output of Algorithm 1 on the lone "no" instance of Unique Search$_{n/2}$ is "yes" with probability α, is $\mathcal{A}(n, \pi)$ for a uniformly random $\pi \in P_0$ with probability $\frac{1-\alpha}{2}$, and is $\neg\mathcal{A}(n, \pi)$ for a

Algorithm 1: An algorithm for UNIQUE SEARCH$_{n/2}$ using $\mathcal{A}$, an algorithm for PERMUTATION$_n$.

Input : An even integer $n \geq 1$.
Oracle : A function $f : [n/2] \to \{0,1\}$ with $\left|f^{-1}(1)\right| \leq 1$.
Output: "yes", if $f^{-1}(1) \neq \emptyset$, "no" otherwise.

With probability $\alpha = \frac{1-2\varepsilon}{3-2\varepsilon}$ **return** *yes* ;

```
// so as to maintain symmetry in the probability of correctness on "yes" and "no"
   instances
```

Implicitly define an "oracle" $h_{\pi,f} : [n] \to [n]$ for some permutation π as follows ;

// We compute $h_{\pi,f}$ only on demand

With probability $\frac{1-\alpha}{2} = \frac{1}{3-2\varepsilon}$ **do**
  Pick a uniformly random permutation $\pi \in P_0$;
  forall the $i \in [n]$ **do**
    if i *is even and* $f(i/2) = 1$ **then** define $h_{\pi,f}(i) = 1$;
    else define $h_{\pi,f}(i) = \pi(i)$; // i is odd, or i is even and $f(i/2) = 0$
  end
  return $\mathcal{A}(n, h_{\pi,f})$;
end

With probability $\frac{1-\alpha}{2} = \frac{1}{3-2\varepsilon}$ **do**
  Pick a uniformly random permutation $\pi \in P_1$;
  forall the $i \in [n]$ **do**
    if i *is odd and* $f((i+1)/2) = 1$ **then** define $h_{\pi,f}(i) = 1$;
    else define $h_{\pi,f}(i) = \pi(i)$; // i is even, or i is odd and $f((i+1)/2) = 0$
  end
  return $\neg\mathcal{A}(n, h_{\pi,f})$;
end

uniformly random $\pi \in P_1$ otherwise. Let ε_0 be the probability of error of $\mathcal{A}$ on a uniformly random oracle from P_0, and let ε_1 be the corresponding quantity for P_1. We have $\varepsilon_0 + \varepsilon_1 = 2\varepsilon$. The probability of error on the "no" instance is

$$\alpha + \left(\frac{1-\alpha}{2}\right)\varepsilon_0 + \left(\frac{1-\alpha}{2}\right)\varepsilon_1 \quad = \quad \alpha + (1-\alpha)\varepsilon \ .$$

The output of Algorithm 1 on a uniformly random "yes" instance of Unique Search$_{n/2}$ is "yes" with probability α, is $\mathcal{A}(n,h)$ for a uniformly random $h \in Q$ with probability $\frac{1-\alpha}{2}$, and is $\neg\mathcal{A}(n,h)$ for a uniformly random $h \in Q$ otherwise. If p denotes the probability that the output $\mathcal{A}(n,h)$ is "no" for a uniformly random $h \in Q$, the probability of error on a uniformly random "yes" instance of Unique Search$_{n/2}$ is

$$\left(\frac{1-\alpha}{2}\right)p + \left(\frac{1-\alpha}{2}\right)(1-p) \quad = \quad \frac{1-\alpha}{2} \ .$$

The choice of $\alpha = \frac{1-2\varepsilon}{3-2\varepsilon}$ makes the two expressions for error probability equal to $\frac{1}{3-2\varepsilon}$.

References

[1] Andris Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64(4):749–898, 2002.

[2] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997.

[3] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing*, pages 212–219. ACM Press, New York, NY, USA, 22–24 May 1996.

[4] Ashwin Nayak and Felix Wu. The quantum query complexity of approximating the median and related statistics. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of computing*, pages 384–393. ACM Press, 1999.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de