INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

# Bounds for Quantum Computational Geometry Problems

*Nilton Volpato* *Arnaldo Moura*

Technical Report - IC-08-07 - Relatório Técnico

April - 2008 - Abril

# Bounds for Quantum Computational Geometry Problems

Nilton Volpato* Arnaldo Moura†

Institute of Computing - University of Campinas (UNICAMP)
Caixa Postal 6176 – 13084-971 – Campinas – SP – Brazil
{nilton,arnaldo}@ic.unicamp.br

**Abstract**

We present new quantum lower bounds and upper bounds for several computational geometry problems. The bounds presented here improve on currently known results in a number of ways. We give asymptotically optimal bounds for one of the problems considered, and we provide up to logarithmic factors optimal bounds for a number of other problems. We settle an open problem of Bahadur et al [4]. Some of these new bounds are obtained using a general algorithm for finding a minimum pair over a given arbitrary order relation.

**Keywords**: quantum computing, quantum algorithm, lower bound, upper bound, computational geometry.

## 1 Introduction

Quantum computing is a new paradigm that is attracting some attention among computer specialists due to evidences that it has some advantages over the classical model of computing. Some of the most widely known quantum algorithms are Shor's polynomial time quantum factoring algorithm [17, 18] and Grover's quantum search algorithm [9, 10]. Both achieve their outstanding complexity performance due to the use of key aspects in quantum computing, such as quantum parallelism and entanglement.

The aim of this paper is to present new lower and upper quantum bounds for several computational geometry problems. We solve some of these problems using a general algorithm for finding a minimum pair over an arbitrarily defined total order relation between pairs of elements, which itself may have other interesting applications for problems in unrelated areas.

Some applications of quantum computing to computational geometry problems were already considered in other works [19, 4, 15]. This paper improves upon these results in a number of ways. Besides describing the algorithms in a more general setting, we provide better upper bounds and better lower bounds to the computational geometry problems

*Research supported by CNPq grant: 140756/2004-3
†Research supported by CNPq grant: 471460/2004-3

considered here. The task of obtaining lower bounds were not even considered in some of the earlier works. The minimum pair algorithm that we develop also settles an open question, posed in [4], that of providing a quantum upper bound for the Furthest Pair problem.

The rest of this article is organized as follows. Section 2 establishes some notation and lists supporting results. In Section 3 we describe and analyse a new algorithm for the Minimum Pair problem. In Section 4 we provide tight bounds for several computational geometry problems using the general algorithm from Section 3 and some *ad hoc* techniques. And we give concluding remarks in Section 5.

## 2 Preliminaries

In this section some supporting algorithms and theorems are listed. Let $N$ be a positive integer. Hereafter we use the following conventions $[N] = \{1, \ldots, N\}$, $\langle N \rangle = (1, \ldots, N)$, similarly, we set $[x_N] = \{x_1, \ldots, x_N\}$ and $\langle x_N \rangle = (x_1, \ldots, x_N)$. If $P$ is a proposition, we define $[\![P]\!] = 1$ if $P$ is true, or $[\![P]\!] = 0$ if $P$ is false.

Throughout the text we use the term *total order relation*, and sometimes just *order relation*, to refer to a relation $\preceq$ that is antisymmetric, transitive and total. When it is not the case that $y \preceq x$ we may simply write $x \prec y$, and when it is the case that $x \preceq y$ and $y \preceq x$ we may write $x = y$.

We say that a function $f : A \to \{0, 1\}$ is non-trivial if and only if there exists some $a \in A$ such that $f(a) = 1$.

### 2.1 Element Distinctness

The Element Distinctness problem has been widely studied both in the classical and the quantum models of computation.

**Definition 1 (Element Distinctness)** *Given $N$ elements $\langle x_N \rangle$ from a given set $X$. Decide if all elements in $\langle x_N \rangle$ are distinct.*

The complexity measure we will use is the total number of oracle queries, or calls. We can use evaluation of comparison oracles. An evaluation oracle $\pi$ is defined as $\pi : [N] \to X$, that is given an index $i$, $\pi(i)$ returns the element $x_i \in X$. We can use, instead, a less powerful comparison oracle, say $\lambda : [N]^2 \to \{0, 1\}$, such an oracle can only tell how the elements whose indices were given compare against each other over an order relation.

In the classical setting $\Theta(N)$ queries to an evaluation oracle are necessary and sufficient for solving the Element Distinctness problem [14]. In the quantum setting $\Theta(N^{2/3})$ queries are necessary [1, 16] and sufficient [3] for solving the Element Distinctness problem, given an evaluation oracle.

If we resort to a comparison oracle, a lower bound of $\Omega(N^{2/3})$ is implied by the lower bound for evaluation queries, and Ambainis [3] showed how to solve the problem using $O(N^{2/3} \log N)$ queries to a comparison oracle.

We summarize the known results in a more general setting.

**Definition 2 (Element $k$-Distinctness)** *Given $N$ elements $\langle x_N \rangle$ from a set $X$, and an integer $k \geq 2$. Decide if there are $k$ identical elements in $\langle x_N \rangle$.*

**Theorem 1 (Element $k$-Distinctness Algorithm [3])** *The Element $k$-distinctness problem can be solved by a quantum algorithm with $O(N^{k/(k+1)})$ queries if using an evaluation oracle and with $O(N^{k/(k+1)} \log N)$ queries if using a comparison one.*

Note that, when $k = 2$, this reduces to the corresponding version of the Element Distinctness problem.

### 2.2 Subset Finding

Building on the work of [3], Childs and Eisenberg [6] pointed out that Ambainis' algorithm actually solved a more general problem, that of finding a subset of size $k$ satisfying some given property. They also gave a somewhat different and simplified analysis of the algorithm. This result may be enunciated as follows.

**Theorem 2 (Subset Finding Algorithm [6])** *Given an integer $k \geq 1$, an evaluation oracle $\pi : [N] \to X$, and a property $P : [N]^k \to \{0, 1\}$. Then, there exists a quantum algorithm that uses $O(N^{k/(k+1)})$ queries to $\pi$, and whose output $[x_k]$ satisfies $P(x_1, \ldots, x_k) = 1$ with constant nonzero probability, if $P$ is non-trivial.*

In this more general setting, we can easily solve the traditional Element Distinctness problem. Just set $k = 2$, $X = [x_N]$, $\pi(i) = x_i$, $P(i, j) = [\![x_i = x_j]\!]$, and apply Theorem 2.

Note that the subset finding algorithm may fail, but it is easy to test if it succeeds by verifying property $P$ over the output. So, by running the algorithm a constant number of times, we can augment the probability of success to a value greater than 1/2.

## 3 The Algorithm for Finding a Minimum Pair

We assume that the set of elements $X$ is a subset of the integers, or equivalently, that there is a bijection from $X$ to a subset of the integers, by this reason we can resort to evaluation oracles.

### 3.1 Threshold Pair Algorithm

We need an auxiliar procedure based on Subset Finding algorithm. Such procedure relies on a total order relation $\preceq$ defined over $X^2$, whose implementation must be given as input. This procedure will find any pair that precedes another given pair over the order relation. We call this given pair as the threshold pair.

Its implementation is straightforward from Theorem 2. By setting $k = 2$, $X = [x_N]$, $\pi(i) = x_i$, and $P(i, j) = [\![(x_i, x_j) \preceq (x_k, x_l)]\!]$. Where $(x_k, x_l)$ is the threshold pair.

Its complexity remain the same as stated by Theorem 2, $O(N^{2/3})$ queries. And we can suppose that its success probability is at least 1/2.

### 3.2 Minimum Pair Algorithm

Using Threshold Pair algorithm, we describe below a quantum algorithm for the Minimum Pair problem. It also uses, as a subroutine, the quantum algorithm of [7], which returns a minimum element among a set of $N$ elements using $O(\sqrt{N})$ queries to an oracle.

**Algorithm: Minimum Pair**

**1.** Choose a random $k \in [N]$. Use the algorithm of [7] to compute $l \in [N]$ such that $(x_k, x_l) \in X^2$ is minimum over the relation $\preceq$.

**2.** Repeat the following $4 \log N$ times.

- **a.** Run the Threshold Pair algorithm using $(x_k, x_l)$ as a threshold pair.
- **b.** Observe its outcome, say $(x_{i'}, x_{j'})$. Fix $i = i'$, and compute, as in step 1, the element $j \in [N]$ such that $(x_i, x_j) \in X^2$ is minimum over the relation $\preceq$.
- **c.** If $(x_i, x_j) \prec (x_k, x_l)$, then set $(x_k, x_l)$ to $(x_i, x_j)$.

**3.** Return $(x_i, x_j)$.

**Lemma 1** *In the Minimum Pair algorithm, suppose that the iteration in step 2 runs for an infinite number of times. Let $X$ be a random variable representing the number of times that the loop needs to be repeated until $(x_k, x_l)$ holds the minimum pair. Then, the expected value of $X$ satisfies* $\mathrm{E}[X] < 2 \log N$.

**Proof.** Consider the set $Z$ of pairs $(k, l)$ where $k$ ranges over $[N]$ and each $l$ is such that $(x_k, x_l)$ is minimum over $\preceq$. We can rank such pairs from 1 to $N$, using $\preceq$ as a base relation.

We want to determine $\mathrm{E}[X]$. For this, let $X_j$ be a random variable that takes on the value 1 if the pair of rank $j$ is chosen at any moment during the algorithm, and 0 otherwise. And let $Y_j$ be a random variable that takes on the number of times that the Threshold Pair algorithm is called, in step 2a, until it succeeds, if it uses the pair with rank $j$ as a threshold.

Then the total number of repetitions of the loop satisfies:

$$X = \sum_{j=2}^{N} X_j Y_j$$

We get the following for the expectation of $X$:

$$\mathrm{E}[X] = \mathrm{E}[\sum_{j=2}^{N} X_j Y_j] = \sum_{j=2}^{N} \mathrm{E}[X_j Y_j].$$

It is easy to see that $X_j$ and $Y_j$ are independent variables, since the probability of success of the Threshold Pair algorithm does not depend on the rank of the threshold. So, we may just write $Y_j = Y$. And since $X_j$ and $Y$ are independent variables, then $\mathrm{E}[X_j Y] = \mathrm{E}[X_j]\,\mathrm{E}[Y]$.

By Lemmas 2 and 3 below, we have:

$$\mathrm{E}[X] = \sum_{j=2}^{N} \mathrm{E}[X_j]\,\mathrm{E}[Y] < \sum_{j=2}^{N} \frac{2}{j} < 2\log N$$

Therefore, $\mathrm{E}[X] < 2\log N$. ■

**Lemma 2** *Let $X_j$ be a random variable as defined in the proof of Lemma 1. Then,* $\mathrm{E}[X_j] = 1/j$.

**Proof.** $X_j$ is a Bernoulli random variable, therefore $\mathrm{E}[X_j] = \Pr(X_j = 1)$. Henceforth we want to prove that $\Pr(X_j = 1) = 1/j$.

Let $X_j^k$ be a random variable that takes on the value 1 if the pair of rank $j$ will ever be chosen when the Threshold Pair algorithm is run using a pair of rank $k+1$ as a threshold, and 0 otherwise. Note that $X_j = X_j^N$ because the first step of the algorithm does not need to be considered as a special case, as the distribution of selecting a pair in step 1 is the same as the output distribution of the Threshold Pair algorithm when using a pair with rank $N+1$ as a threshold.

Clearly, $\Pr(X_j^k = 1) = 0$ if $j > k$ because the values taken by the threshold pair $(k, l)$ are always strictly decreasing due to the test on step 2c.

We will prove that $\Pr(X_j^k = 1) = 1/j$ for each $k \geq j$ and a fixed $j$, by induction on $k$. The basis is $\Pr(X_j^j = 1) = 1/j$ follow due to the fact that the output distribution of the Threshold Pair algorithm is uniform. Now, assume that $\Pr(X_j^l = 1) = 1/j$ for all $j \leq l \leq k$. We want to prove that $\Pr(X_j^{k+1} = 1) = 1/j$. For this case, either the pair with rank $j$ is selected with equal probability from the $k+1$ pairs that have rank less than the threshold, or a pair with rank $i+1$, where $j+1 \leq i+1 \leq k+1$, is selected, and, in a subsequent iteration the pair with rank $j$ is selected. This gives:

$$\Pr(X_j^{k+1} = 1) = \frac{1}{k+1} + \sum_{i=j}^{k} \frac{1}{k+1} \Pr(X_j^i = 1) = \frac{1}{j}.$$

This concludes the proof. ■

**Lemma 3** *Let $Y$ be a random variable as defined in the proof of Lemma 1. Then,* $\mathrm{E}[Y] < 2$.

**Proof.** $Y$ is a geometric random variable with parameter $p$, where $p > 1/2$ because the Threshold Pair algorithm has a probability of success greater than $1/2$. Hence, $\mathrm{E}[Y] = 1/p < 2$. ■

**Lemma 4** *In the Minimum Pair algorithm it suffices to repeat step 2 for $4\log N$ times to achieve a probability of success of at least $1/2$.*

**Proof.** We can use Markov's inequality, which says that for a random variable $Z$ and value $a > 0$, we have $\Pr(Z \geq a) \leq E[Z]/a$. Applying the inequality for the variable $X$ from Lemma 1, we get the desired result. ■

**Theorem 3 (Minimum Pair)** *Given a total order relation $\preceq \subseteq X^2 \times X^2$. Then, there exists a quantum algorithm that outputs $(x_1, x_2)$, such that $(x_1, x_2)$ is the minimum over the relation $\preceq$. Furthermore, such an algorithm uses $O(N^{2/3} \log N)$ queries of an evaluation oracle $\pi$.*

**Proof.** Lemma 4 shows that the number of repetitions used in step 2 suffices, and that the algorithm has at least 1/2 probability of being successful.

It remains to prove the complexity bounds. Steps 1 and 2b require $O(\sqrt{N})$ queries, step 2a requires $O(N^{2/3})$ queries due to the Threshold Pair algorithm, and step c requires no oracle query. Since step 2 is repeated $4 \log N$ times, this gives $O(N^{2/3} \log N)$ overall queries. ■

# 4 Tight Bounds for Some Computational Geometry

In this section we revisit some problems in computational geometry. The Closest, Furthest, Closest Bichromatic and Furthest Bichromatic Pair problems considered in Sections 4.1, 4.2 and 4.3 all use the general algorithm from Section 3. For the Smallest Enclosing Ball and Segment Intersection Detection problems, in Sections 4.4 and 4.5, we present lower bounds and upper bounds using specific techniques for these problems.

## 4.1 Closest Pair Problem

The closest pair problem is defined as follows.

**Definition 3 (Closest Pair)** *Given $N$ points $P = [p_N]$ in $d$-dimensional space and a distance function $d : P \times P \to \mathbb{R}$, find a pair of points which are closest to each other.*

In the classical deterministic computational model, $\Theta(N \log N)$ comparisons or $\Theta(N)$ evaluations are necessary and sufficient [14]. In the classical probabilistic model, $O(N)$ comparisons suffice [8, 12]. In the quantum computational model, $\Omega(N^{2/3})$ queries are necessary for solving the closest pair problem [19, 4].

We can solve this problem using the algorithm from Theorem 3. To do this, we set $X = P$ and we define the order relation $(p_1, p_2) \preceq (q_1, q_2)$ if and only if $d(p_1, p_2) \leq d(q_1, q_2)$, where $d$ is the given distance function. Using the algorithm we get a bound of $O(N^{2/3} \log N)$ queries for solving the closest pair problem in any number of dimensions and using any distance function.

For 2 dimensions and using the Euclidean metric $L_2$ an $O(N^{2/3} \log N)$ quantum algorithm was proposed in [4]. For $d \geq 3$ dimensions, the best previous known upper bound was $\tilde{O}(N^{1-1/4\lceil d/2 \rceil})$, due to [15].

Our proposal is more general than the previous ones, and it works for any number of dimensions. It outperforms, or ties up to, all previous algorithms. The only tie up case being when compared to the algorithm from [4], that works only for 2 dimensions and uses the Euclidean metric.

### 4.2 Furthest Pair Problem

The similar furthest pair problem, or diameter problem, is defined as follows.

**Definition 4 (Furthest Pair)** *Given $N$ points $P = [p_N]$ in $d$-dimensional space and a distance function $d : P \times P \to \mathbb{R}$, find a pair of points which are furthest to each other.*

Classically, the Closest Pair problem and the Furthest Pair problem have the same complexity, with $\Theta(N \log N)$ comparisons or $\Theta(N)$ evaluations being necessary and sufficient [14]. In the quantum setting, the best known lower bound is $\Omega(N^{2/3} \log^{-1/2} N)$ queries, due to [4].

The best so far upper bound to this problem was $O(N^{3/4} \log^2 N)$ queries for $d = 2$ and $\tilde{O}(N^{1-1/4\lceil d/2 \rceil})$ queries for $d \geq 3$, due to [15]. Our algorithm outperforms this in all cases, using $O(N^{2/3} \log N)$ evaluation queries for any number of dimensions.

To solve this problem, we apply again Theorem 3, as was done for the Closest Pair problem in Section 4.1. Just set $X = P$ and define the order relation $(p_1, p_2) \preceq (q_1, q_2)$ if and only if $d(p_1, p_2) \geq d(q_1, q_2)$, where $d$ is the given distance function. The result, as already stated, is an algorithm whose number of oracle queries is $O(N^{2/3} \log N)$.

### 4.3 Closest Bichromatic Pair Problem

**Definition 5 (Closest Bichromatic Pair Problem)** *Given $N$ points $P = [p_N]$ and $M$ points $Q = [q_M]$ in $d$-dimensional space, and a distance function $d : (P \cup Q) \times (P \cup Q) \to \mathbb{R}$. Find a pair of points $(p, q) \in P \times Q$ which are closest to each other.*

In the classical probabilistic model, an algorithm is presented in [2] for solving the Closest Bichromatic Pair problem with an $O((NM \log N \log M)^{2/3} + M \log^2 N + N \log^2 M)$ time complexity.

In the quantum setting, if we let $L = M + N$, then a $\Omega(L^{2/3})$ quantum lower bound is implied since this problem is more general than the Closest Pair problem. For the upper bound, the best bounds known so far were $O(L^{3/4} \log L)$ for $d = 2$ and $\tilde{O}(L^{1-1/4\lceil d/2 \rceil})$ for $d \geq 3$, both due to [15].

We can also use our algorithm to solve this problem in $O(L^{2/3} \log L)$ queries of an evaluation oracle. To do this, we set $X = P \cup Q$ and define $\pi : [N + M] \to X$ with $\pi(i) = p_i$ for $1 \leq i \leq N$ and $\pi(N + i) = q_i$ for $1 \leq i \leq M$. Further, we define the order relation $\preceq$, as in Section 4.1, by letting $(x_1, x_2) \preceq (y_1, y_2)$ if and only if $d'(x_1, x_2) \preceq d'(y_1, y_2)$. Where $d'$ is defined as $d'(x, y) = d(x, y)$ if $(x, y) \in (P \times Q) \cup (Q \times P)$, and $d'(x, y) = \infty$ otherwise. Now apply Theorem 3. Clearly, a successful run of the Minimun Pair algorithm will only produce pairs from $(P \times Q) \cup (Q \times P)$, thus solving the Bichromatic Pair problem.

In a similar vein, we can solve the Furthest Bichromatic Pair problem within the same complexity bounds.

### 4.4 Smallest Enclosing Ball

The Smallest Enclosing Ball problem, also known as Smallest Enclosing Circle problem when the points lie on the plane, is defined as follows.

**Definition 6 (Smallest Enclosing Ball)** *Given $N$ points $P = [p_N]$ in $d$-dimensional space, find the smallest $d$-dimensional sphere that encloses them.*

To obtain a lower bound for this problem, we present a reduction from the OR problem, which is defined below.

**Definition 7 (OR)** *Given a set $X = [x_N]$ of $N$ binary variables decide if at least one of them is equal to* 1.

**Theorem 4** *The quantum query complexity of the Smallest Enclosing Ball problem is* $\Omega(N^{1/2})$.

**Proof.** It is a known fact that the OR problem has a lower bound of $\Omega(N^{1/2})$ queries in the quantum bounded-error model [5, 11, 13]. We proceed by reducing the OR problem to the Smallest Enclosing Ball problem in 2 dimensions.

We are given an instance $X = [x_N]$ of the OR problem, which we convert to an instance $P = [p_{N+2}]$ of the Smallest Enclosing Circle problem with $N+2$ points, just let $p_1 = (1, 0)$, $p_i = (i, (N + 2)x_{i-1})$ for $2 \leq i \leq N + 1$, and $p_{N+2} = (N + 2, 0)$.

More formally, suppose that we have an oracle $\pi : [N] \rightarrow [x_N]$ for the OR problem. We construct an oracle $\pi'$ for the Smallest Enclosing Circle problem by letting $\pi' : [N + 2] \rightarrow [p_{N+2}]$, where $\pi'(1) = (1, 0)$, $\pi'(i) = (i, (N + 2)\pi(i - 1))$ for $2 \leq i \leq N + 1$, and $\pi'(N + 2) = (N + 2, 0)$. By doing so, each call to $\pi'$ incurs in one call to $\pi$.

It is not difficult to see that, for this configuration, the OR problem has answer "no" if and only if the Smallest Enclosing Circle has a radius of length exactly $(N + 1)/2$. ■

The previous known upper bound for this problem, due to Sadakane et al. in [15], was $O(\sqrt{N} \min\{\log^d \Gamma \log N, \log^{2d+1} N\})$, where $\Gamma$ is some given precision for the input, and $d$ is the dimension.

### 4.5 Segment Intersection Detection

This problem is defined as follows.

**Definition 8 (Segment Intersection Detection)** *Given a set $S$ of $N$ line segments in the plane, detect whether $S$ has a pair of intersecting segments.*

To give a lower bound for this problem, we give a reduction from the Element Distinctness problem of Section 2.1.

**Theorem 5** *The quantum query complexity of the Segment Intersection Detection problem is* $\Omega(N^{2/3})$.

**Proof.** In a simple reduction, each element $x_i$ from the Element Distinctness problem is mapped to a line segment from $(0, x_i)$ to $(i, x_i)$. It's easy to see that the segments will intersect if and only if there is a non unique element. ■

The previous known upper bound for this problem was $O(N^{7/8})$ by [15]. We can readily adapt the element distinctness algorithm for solving this problem in the following way.

**Theorem 6** *The Segment Intersection Detection problem can be solved using* $O(N^{2/3})$ *queries in the quantum model.*

**Proof.** The algorithm for Element Distinctness can be adapted to solve the Segment Intersection Detection problem. For this, we use an oracle $\pi : [N] \to S$, where $S$ is the set of given line segments in the plane. Furthermore, we need to replace the equality test by a line segment intersection test. Testing whether two line segments intersects takes only constant time and no queries, since the segments are already determined by the call to the oracle $\pi$. The Element Distinctness algorithm will detect a "non unique" element if and only if there is a segment intersection in $S$. The details easily follow. ∎

## 5 Conclusions

The minimum pair algorithm we presented is optimal for the problems considered here, up to logarithmic factors.

Our algorithm outperformed all previous algorithms for the problems being considered. When the Closest Pair problem is being solved for the particular case of the Euclidean metric in 2 dimensions, our algorithm has the same complexity as that of [4]. However our algorithm is presented in a much more general setting, which also accounts for dealing with other metrics and any number of dimensions.

In Section 4.2 we have also settled the open problem proposed by [4], which asked for an upper bound for the query complexity of the Furthest Pair problem.

As for the Smallest Enclosing Ball, we have provided a new better lower bound. For the Segment Intersection Detection problem we improved on previous results, providing new matching lower and upper bounds, thereby providing an asymptotically optimal algorithm for that problem.

Finally, we believe that a generalization of the algorithm presented here for finding a Minimum Pair to an algorithm for finding a Minimum Tuple—instead of just a pair—may have some interesting applications in solving problems in computational geometry and in other areas as well.

## Summary of the Results

| Problem | Earlier works | Present work |
|---|---|---|
| Closest Pair | $\Omega(N^{2/3})$ [19, 4], $O(N^{2/3}\log N)^{\dagger}$ [4] | $\Omega(N^{2/3})$, $O(N^{2/3}\log N)$ |
| Furthest Pair | $\Omega(N^{2/3}\log^{-1/2} N)$ [4] | $O(N^{2/3}\log N)^{\ddagger}$ |
| Closest Bichromatic Pair | $O(L^{3/4}\log^2 L)$ for $d = 2$ [15]<br>$\tilde{O}(L^{1-1/4\lceil d/2\rceil})$ for $d > 2$ [15] | $\Omega(L^{2/3}), O(L^{2/3}\log L)$ |
| Smallest Enclosing Ball | $\tilde{O}(\sqrt{N})^{\S}$ [15] | $\Omega(N^{1/2})$ |
| Segment Intersection | $\tilde{O}(N^{7/8})$ [15] | $\Theta(N^{2/3})$ |

†Works only for the Euclidean $L^2$ metric and 2 dimensions.

‡Closes an open problem posed in [4].

§The complexity presented in [15] is, in fact, $O(\sqrt{N}\min\{log^d\Gamma\log N, \log^{2d+1} N\})$, which takes into consideration the precision $\Gamma$ of the input.

## References


[1] Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51:595–605, 2004.

[2] Pankaj K. Agarwal, Herbert Edelsbrunner, Otfried Schwarzkopf, and Emo Welzl. Euclidean minimum spanning trees and bichromatic closest pairs. In *SCG '90: Proceedings of the sixth annual symposium on Computational geometry*, pages 203–210, New York, NY, USA, 1990. ACM Press.

[3] Andris Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37:210–239, 2007. http://arxiv.org/abs/quant-ph/0311001. Also on FOCS'04.

[4] Abhinav Bahadur, Christoph Dürr, Raghav Kulkarni, and Thibauly Lafaye. Quantum query complexity in computational geometry. In *Proc. of the Conference on Quantum Information and Computation IV by The International Society for Optical Engineering (SPIE)*, volume 6244, page 624413, 2006.

[5] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. In *39th Anual Symposium on Foundations of Computer Science (FOCS)*, pages 352–361, 1998. http://arxiv.org/abs/quant-ph/9802049.

[6] Andrew M. Childs and Jason M. Eisenberg. Quantum algorithms for subset finding. *Quantum Information And Computation*, 5:593, 2005. http://arxiv.org/abs/quant-ph/0311038.

[7] Christoph Dürr and Peter Hoyer. A quantum algorithm for finding the minimum, 1996. http://arxiv.org/abs/quant-ph/9607014.

[8] Mordecai Golin, Rajeev Raman, Christian Schwarz, and Michiel Smid. Simple randomized algorithms for closest pair problems. *Nordic J. of Computing*, 2(1):3–27, 1995.

[9] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proc. 28th Annual ACM Symposium on the Theory of Computation*, pages 212–219, New York, NY, 1996. ACM Press, New York. `http://arxiv.org/abs/quant-ph/9605043`.

[10] Lov K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79:325, 1997. `http://arxiv.org/abs/quant-ph/9706033`.

[11] Peter Hoyer and Robert Spalek. Lower bounds on quantum query complexity. *EATCS Bulletin*, 87:78–103, October 2005. `http://arxiv.org/abs/quant-ph/0509153`.

[12] Samir Khuller and Yossi Matias. A simple randomized sieve algorithm for the closest-pair problem. *Inf. Comput.*, 118(1):34–37, 1995.

[13] Michael L. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[14] Franco P. Preparata and Michael Ian Shamos. *Computational Geometry an introduction*. Springer-Verlag New York Inc., 1985.

[15] Kunihiko Sadakane, Norito Sugawara, and Takeshi Tokuyama. Quantum computation in computational geometry. *Interdisciplinary Information Sciences*, 8(2):129–136, 2002.

[16] Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems, 2001. `http://arxiv.org/abs/quant-ph/0112086`.

[17] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In S. Goldwasser, editor, *Proceedings of the 35th Annual Symposium on the Foundations of Computer Science*, pages 124–134, Los Alamitos, CA, 1994. IEEE Computer Society.

[18] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J.Sci.Statist.Comput.*, 26:1484, 1997. `http://arxiv.org/abs/quant-ph/9508027`.

[19] Nilton Volpato and Arnaldo Moura. An $O(N^{3/4}\log^2 N)$ quantum algorithm for the 1-dimensional closest-pair problem. In *WECIQ 2006: Workshop-Escola de Computação e Informação Quântica*, Pelotas, Brazil, October 2006.