Chapter on Quantum for
Computational Intractability:
A Guide to Algorithmic Lower Bounds
by Demaine, Gasarch, Hajiaghayi

Note to Proofreaders There will be passages like "In Chapter ?? we discussed..." This is not a mistake. This is a a result of giving you this chapter and not the rest of the book. We have not put it into the book yet.

1 Introduction

This book is about classical computing. The algorithms and reductions in this book can be carried out by a modern computer running on a single CPU or multiple CPUs. We will call such computers ***classical*** in that they are based on classical physics (at the bit level computers are based on electricity) and not quantum physics. We use the term ***classical algorithm*** for an algorithm that can be run on a classical computer. What we call a ***classical algorithm*** in this chapter was just an ***algorithm*** in all of the other chapters.

There are theoretical devices called ***quantum computers***. An algorithm that can be run on such a device is a ***quantum algorithm***. There are some problems for which, theoretically, there is a quantum algorithm that is faster than any (known) classical algorithm. Hence quantum algorithms are of interest (we later list other reasons they are of interest).

Quantum computing is a vast topic that we will, for reasons of space, only be able to discuss briefly. We will have very few definitions, proofs, algorithms, or reductions. For basic definitions and more information on quantum computing see (1) the references in this chapter, (2) many websites that you can get to by doing a web search on `Quantum Computing`, and (3) the books by Aaronson [Aar13], Mermin [Mer07], or Nielson & Chuang [NC16].

In this chapter we will state results about quantum algorithms and, in some cases, compare them to classical algorithms. We will also look at a quantum version of a classical problem. The topics and results chosen highlight when the classical world and the quantum world differ or seem to differ.

A cautionary note: from the popular press one might think that quantum computers, if they are built, can solve world hunger, predict the stock market, and solve **NP**-complete problems. Most experts agree that this is not the case in reality. This misunderstanding may come from a misinterpretation of the *many-worlds interpretation of quantum mechanics* which gives the false impression that quantum algorithms are massively parallel. They are not. In reality there seem to be only a few problems of interest that quantum computers (if they are built) can do much faster than classical computers. We also note that most experts in quantum do not think that **NP**-complete problems can be solved quickly by a quantum computer.

Given that the usefulness of quantum algorithms seems limited, why study them?

1. There are two problems, *Factoring* and *Discrete Log*, which (1) are very important, and (2) quantum algorithms for them are much faster (polynomial time versus exponential time) than any known classical algorithm.

2. There are many problems that have quantum algorithms that are faster than any known classical algorithm. Childs & Dam [CvD10] survey algebraic problems that have quantum algorithms which seem faster than any known classical algorithm. Jordan [Jor11] maintains a website of problems that have quantum algorithms that seem faster than any known classical

algorithm. The speedups are usually not that large. Even so, they are a proof-of-concept for quantum algorithms being useful.

3. Richard Feynman first conceived of quantum computing as a way to potentially simulate quantum mechanics. This is another problem where quantum computers may outperform classical ones.

4. There have been cases where a classical algorithm was inspired by research on quantum algorithms. We give an example. Kerenidis & Prakash [KP17b] had a quantum algorithm for a recommendation system. Tang, while trying to show that no classical algorithm could do as well as the quantum algorithm, found a classical one that did [Tan19]. We stress that this classical algorithm was found because of research in quantum algorithms. For other examples do a websearch for `Quantum Inspired Classical Algorithms`.

5. The study of quantum algorithms has lead to results in classical computing. See the survey of Drucker & de Wolf [DdW11] for examples.

6. The attempt to build quantum computers may lead to interesting insights into quantum physics. See next point.

7. The most exciting development that could happen would be if the attempt to build quantum computers leads to a discovery that the current theories of quantum mechanics are wrong or incomplete.

2 Factoring

Factoring is an important problem for cryptography. Many cryptosystems would be broken if factoring is easy. Hence, in contrast to work in algorithms, cryptographers hope that factoring is hard.

We will define factoring slightly differently than how it was defined in Chapter **??**.

2.1 Classical Factoring

Problem 2.1. FACTORING *(*FACT*)*
INSTANCE: A number N
QUESTION: If N is prime then output **PRIME***. If N is not prime then output a non-trivial factor of N.*

As noted in Chapter **??** there are no polynomial-time algorithms for FACT, nor is there a proof that its **NP**-complete. There are reasons to think it is not **NP**-complete (See Exercise **??**).

Algorithms for factoring are hard to analyze and depend on (widely believed) conjectures in number theory. The fastest known algorithm, the General Number Field Sieve, is believed to have running time roughly $2^{1.93\,L^{1/3}(\lg L)^{2/3}}$, where $L = \lg N$ is the length of the input number N. This bound is small enough that the algorithm is practical for moderately large inputs. The naive algorithm for factoring takes time $2^{L/2}$. Hence reducing the time to roughly $2^{L^{1/3}}$ is a real improvement. The first one to come up with the idea for the Number Field Sieve is Pollard, though his work was never published. See the survey of Pomerance [Pom96] or the collection of articles on the Number field Sieve edited by A. Lenstra & H. Lenstra [LL93].

There is no clear consensus on whether FACT is in **P**: in Gasarch's [Gas19] 2019 Poll on **P** vs **NP** he also asked about FACT. Of the 108 people who responded 38 (35%) thought FACT $\in$ **P**, while 70 (65%) thought FACT $\notin$ **P**. It's been said that cryptographers think (hope?) that FACT $\notin$ **P** while number theorists think FACT $\in$ **P**.

If FACT $\in$ **P** this will require new techniques. Here is why:

1. The last improvement in factoring algorithms was the Number Field Sieve in 1988.
2. There are reasons to think that the current methods yield algorithms with running times of the form $2^{L^t(\ln L)^{1-t}}$, where $0 < t < 1$. The General Number Field Sieve achieves $t = 1/3$. It is plausible that current techniques will solve FACT in time (say) $2^{L^{1/10}(\ln L)^{9/10}}$ but not in **P**.

For more about factoring, see Wagstaff's book [Wag13].

2.2 Quantum Factoring

What about Quantum Polynomial time?

Theorem 1. *Let the input to a factoring algorithm be N. Let $L = \lg N$ which is the length of N.*

1. *(Shor [Sho94, Sho99]) There is a polynomial quantum algorithm for* FACT*.*
2. *(Beckman et al. [BCDP96]) There is quantum algorithm for* FACT *that takes time $O(L^2 \log(L) \log(\log(L)))$. (This paper used Shor's algorithm as a starting point.)*

We note the following

1. The key quantum component of Shor's algorithm for FACT is the quantum Fourier transform.
2. The constants in Beckman et al.'s version of Shor's algorithm are small. The biggest obstacle to running the algorithm is building a quantum computer that can handle many qubits.
3. Martin-Lopez et al. [MLLL$^+$12] have factored 21 on a quantum computer using Shor's algorithm. This can be considered a proof-of-concept. Other bigger numbers have been reported to have been factored by *a quantum computer* but they really used a lot of classical computing to set the problem up and hence we do not count those. Smolin et al. [SSV13] discuss this issue.

Upshot If FACT $\notin$ **P** then FACT will be an example of a problem that quantum computers can do faster than classical computers. Proving FACT $\notin$ **P** is hard since it implies **P** $\neq$ **NP**.

3 Discrete Log

Discrete Log is an important problem for cryptography. Many cryptosystems would be broken if Discrete Log is easy. Hence, in contrast to work in algorithms, cryptographers hope that Discrete Log is hard.

We will define Discrete Log slightly differently than how it was defined in Chapter **??**.

3.1 Classical Discrete Log

Problem 3.1. Discrete Log *(*DL*)*

INSTANCE: A prime p, a generator g of $\mathbb{Z}_p$, and $a \in \mathbb{Z}_p$. (g is a generator if $\{g, g^2, \ldots, g^{p-1}\} = \{1, \ldots, p-1\}$.)

QUESTION: Find x such that $g^x \equiv a \pmod{p}$.

As noted in Chapter **??** there are no polynomial-time algorithms for DL, nor is there a proof that its **NP**-complete. There are reasons to think it is not **NP**-complete (See the discussion of DL in Chapter **??**.)

Algorithms for DL are hard to analyze and depend on (widely believed) conjectures in number theory. The fastest known algorithm, the Function Field Sieve, is believed to have running time roughly $2^{1.53\,L^{1/3}(\lg L)^{2/3}}$, where $L = \lg N$ is the length of the input number N. This bound is small enough that the algorithm is practical for moderately large inputs. The naive algorithm for DL takes time 2^L. Hence reducing the time to roughly $2^{L^{1/3}}$ is a real improvement. Adleman [Adl94] developed the Function Field Sieve and then elaborated the ideas with Huang (see [AH99]).

There is no formal connection between DL and FACT; however, the techniques for one seem to apply to the other. Hence the following two points made about FACT are true of DL also: (1) there is no consensus about if DL $\in$ **P**, and (2) if DL $\in$ **P** then this will require new techniques.

3.2 Quantum Discrete Log

What about Quantum Polynomial time?

Theorem 2. *Let the input to a* DL *algorithm be N. Let $L = \lg N$ which is the length of N.*

1. *(Shor [Sho94, Sho99]) There is a polynomial quantum algorithm for* DL*.*

2. *(Folklore though can be obtained from Beckman et al. [BCDP96].) There is quantum algorithm for* DL *that takes time $O(L^2 \log(L) \log(\log(L)))$.*

We note the following

1. The key quantum component of Shor's algorithm for DL is the quantum Fourier transform. In Section 3.1 we noted that while there is no formal connection between DL and FACT, improvements in one tend to lead to improvements in the other. We were referring to classical algorithms. However, the same seems to be true for quantum algorithms: both the quantum algorithm for FACT and for DL use the quantum Fourier transform.

2. There do not seem to be any attempts to execute Shor's DL algorithm on a quantum computer. Since the quantum algorithms for FACT and DL are similar, the same techniques that were used for FACT will work on DL. Hence it is likely that a quantum computer could be used to find DL when p, g, a are all ≤ 21.

Upshot If DL $\notin$ **P** then DL will be an example of a problem that quantum computers can do much faster than classical computers. Proving DL $\notin$ **P** is hard since it implies **P** $\neq$ **NP**.

4 The Search Problem

Problem 4.1. SEARCH

INSTANCE: Access to a function $f\colon \{0,\dots,N-1\} \to \{0,1\}$. We are promised that there is only one x such that $f(x)=1$. We think of this function as representing a 1-element subset of $\{0,\dots,N-1\}$.

QUESTION: Find the x such that $f(x)=1$.

NOTE: The basic unit of computation is an evaluation of f which we call a query.

Theorem 3. *Let $N \in \mathbb{N}$ and $f\colon \{0,\dots,N-1\} \to \{0,1\}$.*

1. *(Easy) There is a deterministic algorithm for* SEARCH *that takes N queries in the worst case. There is a randomized algorithm for* SEARCH *that has expected complexity $\frac{N}{2}$ queries. Both of these are optimal.*
2. *(Grover [Gro96]) There is a quantum algorithm for* SEARCH *that uses $O(\sqrt{N})$ queries.*
3. *(Bennett et al. [BBBV97]) Any quantum algorithm for* SEARCH *requires $\Omega(\sqrt{N})$ queries.*
4. *(easy) Assume that instead of having only 1 x with $f(x)=1$ there are M, and the goal is to find one of them. There is a deterministic algorithm takes $O(N-M)$ queries in the worst case. There is a randomized algorithm that has expected complexity $\frac{N}{M}$ queries. Both of these are optimal.*
5. *For the problem in the last item there is a quantum algorithm that takes $O(\sqrt{N/M})$ queries.*

Upshot SEARCH, with the complexity measure number-of-queries, is a problem where quantum computers are **provably** faster than classical computers.

5 The Traversal Problem

Problem 5.1. TRAVERSAL

INSTANCE: A graph G with $\Theta(2^n)$ vertices represented by $\Theta(n)$-bit strings. There are two distinguished vertices ENTRANCE *and* EXIT*. The label of* ENTRANCE *(e.g., Vertex 0110) is given. The label of* EXIT *is not given.*

QUESTION: Find the label of EXIT*.*

NOTE: The graph is large. We think of the graph as being like a database that you ask questions about. The algorithm can ask question of the following type: given a string w of $\Theta(n)$ bits, return the following information:

- *Is w a vertex?*
- *If w is a vertex then output all of its neighbors.*
- *If one of the neighbors of w is* EXIT *then indicate this.*

The number of queries is the complexity of the algorithm.

NOTE: TRAVERSAL*is only asking to find* EXIT*. It is not asking to find the path from* ENTRANCE *to* EXIT*. We will later comment on this perhaps harder problem of having to find the path.*

NOTE: The basic unit of computation is a query of one of the types above.

This problem looks like it requires $\Omega(2^n)$ queries for either classical or quantum algorithms. And indeed, for the case of general graphs, that is the case. But there is a sequence of graphs where there is a large difference between classical algorithms and quantum algorithms.

One technique that a classical algorithm can use is a CLASSICAL RANDOM WALK: the algorithm picks a random neighbor of ENTRANCE, then a random neighbor of that neighbor, etc, until it finds EXIT. There is also a notion of a QUANTUM RANDOM WALK which we will not define.

Theorem 4.

1. *(Childs et al. [CFG02]) There is a sequence of graphs* $\{G_n\}_{n=1}^{\infty}$ *such that the following hold: (a)* G_n *has* $\Theta(2^n)$ *vertices, (b) there is a* QUANTUM RANDOM WALK *algorithm that solves* TRAVERSAL *on* G_n *using* $\le p(n)$ *queries for some polynomial* p*, (c) any* CLASSICAL RANDOM WALK *algorithms requires* $2^{\Omega(n)}$ *queries.*

2. *(Childs et al. [CCD+03]) There is a sequence of graphs* $\{G_n\}_{n=1}^{\infty}$ *such that the following hold: (a)* G_n *has* $\Theta(2^n)$ *vertices, (b) there is a* QUANTUM RANDOM WALK *algorithm that solves* TRAVERSAL *on* G_n *with* $p(n)$ *queries for some polynomial* p*, (c) any classical algorithms (whether or not it uses* CLASSICAL RANDOM WALK*) requires* $2^{\Omega(n)}$ *queries.*

3. *(Jeffery and Zur [JZ22]) In items 1 and 2,* $p(n)$ *can be replaced by* $O(n)$*.*

4. *(Childs et al. [CCG23]) (Informal) Consider the problem of actually finding the path from* ENTRANCE *to* EXIT*. Under reasonable assumptions, any* CLASSICAL RANDOM WALK *or* QUANTUM RANDOM WALK *algorithm requires an exponential number of queries.*

Upshot TRAVERSAL, with the complexity measure number-of-queries, is a problem where quantum computers are **provably** faster than classical computers.

6 Subquadratic Approximate Edit Distance

Definition 1. *Let* Σ *be a finite alphabet and let* $x, y \in \Sigma^*$*. The* ***edit distance between x and y*** *is the number of insertions/deletions/substitutions needed to transform* x *into* y*.*

Problem 6.1. EDIT DISTANCE
INSTANCE: Two strings x, y *over some alphabet* Σ*. We think of* Σ *as being fixed.*
QUESTION: What is the edit distance between x *and* y*?*

Theorem 5.

1. *(Easy)* EDIT DISTANCE *can be computed in time* $O(n^2)$ *where* $n = \max\{|x|, |y|\}$*.*

2. *(Backurs & Indyk [BI15]) Assuming* ***SETH****,* EDIT DISTANCE *requires* $\Omega(n^2)$ *time.*

3. *(Abboud et al. [AHWW16]) With an assumption weaker than* ***SETH****,* EDIT DISTANCE *requires* $\Omega(n^2)$ *time.*

Theorem 5 settles the question for ***exact*** EDIT DISTANCE: quadratic time is both the upper and lower bound. Is there a subquadratic algorithm for approximating EDIT DISTANCE?

Can a quantum algorithm give a subquadratic approximation algorithm? Yes. Boroujeni et al. [BEG+21] proved the following:

Theorem 6.

1. *(Theorem 4.5 of their paper) For all $\epsilon > 0$ there is a quantum algorithm that (a) runs in time $O(n^{2-(4/21)}\log(\frac{1}{\epsilon}))$ and (b) returns a number that is $\le (3+\epsilon)\mathrm{OPT}(x,y)$. Note that $2-(4/21) \approx 1.8095$.*

2. *(Theorem 5.1 of their paper) For all $\epsilon > 0$ there is a quantum algorithm that (a) runs in time $\tilde{O}(n^{\alpha})$ where $\alpha = 2-(5-\sqrt{17}/3) \approx 1.7077$. (b) returns a number that is $\le O(1/\epsilon)^{O(1/\epsilon)\mathrm{OPT}(x,y)}$.*

So at this point it looks like quantum computers can give a constant approximation but classical can not. But then Chakraborty [CDG+20] obtained a constant approximation by taking one of the steps of the quantum algorithm of Boroujeni et al. [BEG+21] and figuring out how to do it classically. This is discussed in both Chakraborty [CDG+20] and a blog post of Rubinstein [Rub18]. Chakraborty [CDG+20] showed the following:

Theorem 7. *There exists a constant C and a randomized algorithm that (a) runs in time $\tilde{O}(n^{2-(2/7)})$ and (b) with probability $1-n^{-5}$ returns a number that is $\le C\mathrm{OPT}(x,y)$. Note that $2-(2/7) \approx 1.7142$. We note that the constant C is large.*

Andoni & Nosatzki [AN20] obtained a classical subquadratic approximation result that is parameterized by ϵ.

Theorem 8. *For all $\epsilon > 0$ there is an algorithm that (a) runs in time $O(n^{1+\epsilon})$ and (b) returns a number that is $\le f(\frac{1}{\epsilon})\mathrm{OPT}(x,y)$ where f is not given explicitly but is roughly double exponential in $\frac{1}{\epsilon}$.*

Open 1. *In this open problem the problem is of course approximate* Edit Distance.

1. *Find a constant $D > 3$ such that that no classical algorithm can, in subquadratic time, obtain a $D\mathrm{OPT}(x,y)$ approximation. This would show that a subquadratic $(3+\epsilon)$-approximation for* Edit Distance *is a problem that a quantum algorithm can do but a classical one cannot.*

2. *Improve the runtime of the quantum algorithm in Theorem 6.1.*

TWO UPSHOTS: The problem at hand is Edit Distance.

1. The following can be done by a quantum algorithm but (at least for now) not by a classical algorithm: a subquadratic algorithm that, on input x, y, returns $(3+\epsilon)\mathrm{OPT}(x,y)$.

2. For this problem a quantum approximation algorithm inspired a classical one.

7 Quantum Streaming Algorithms

7.1 Classical Streaming for Triangle Counting and Distinguishing

Problem 7.1. Triangle Counting TC
INSTANCE: Graph $G = (V, E)$
QUESTION: Approximate the number of triangles in G.

A related problem that is usually considered in the literature is that of Triangle Distinguishing, which is defined as follows.

Problem 7.2. TRIANGLE DISTINGUISHING TD

INSTANCE: Graph $G = (V, E)$, a number T, and the promise that G has either 0 triangles or T triangles.

QUESTION: Does G have 0 triangles or T triangles?

Clearly TD $\leq$ TC. Hence, a lower bound on TD implies a lower bound on TC.

We will state lower bounds on TD (and hence TC). Recall that in Chapter **??** lower bounds on streaming algorithms were obtained via lower bounds in communication complexity. In this chapter we will talk about (though not prove) lower bounds on quantum streaming algorithms via lower bounds on quantum communication complexity. We first need a problem with large quantum communication complexity.

Definition 2. *Let $n \in \mathbb{N}$.*

1. *A **perfect matching M over $[2n]$** is a set of n ordered pairs (i, j), where i and j are distinct elements of $[2n]$, such that every $\ell \in [2n]$ is in exactly 1 ordered pair.*
2. *Let M be a perfect matching over $[2n]$. We identify M with the following $n \times 2n$ matrix: For every ordered pair (i, j) in the matching there is a row with 1's in the ith and jth spot, and 0's everywhere else. Note that a perfect matching can be associated to many different matrices. We will turn this around: we will give Bob a perfect matching by giving him a matrix.*

Problem 7.3. BOOLEAN HIDDEN MATCHING BHM

INSTANCE: Alice gets a string $x \in \{0, 1\}^{2n}$. Bob gets (a) a perfect matching M over $[2n]$ via a matrix as described in Definition 2, and (b) a string $w \in \{0, 1\}^n$ where w is promised to satisfy either $Mx = w$ or $Mx = \overline{w}$ (where $\overline{w}$ is w with every bit flipped).

QUESTION: Determine which is the case: $Mx = w$ or $Mx = \overline{w}$.

Theorem 9. *(Gavinsky et al. [GKK+08]) The randomized 1-way communication complexity of* BHM*, with Alice sending, is $\Omega(\sqrt{n})$.*

Notation 1. *Let n denote the number of vertices, m denote the number of edges, and T is as in the problem statement. Δ_V (respectively Δ_E) is the maximum number of triangles in G that share a vertex (respectively an edge).*

The following are known.

Theorem 10.

1. *(Jayaram & Kallaugher [JK21]) There is a single-pass streaming algorithm for* TC *that uses space $\tilde{O}\left(\frac{m\Delta_E}{T} + \frac{m\sqrt{\Delta_V}}{T}\right)$.*
2. *(Braverman et al. [BOV13]) Any single-pass streaming algorithm for* TD *(and hence for* TC*) uses space $\Omega\left(\frac{m\Delta_E}{T}\right)$. This proof uses a reduction of* INDEX *to* TD.
3. *(Kallaugher and Price [KP17a]) Any single-pass streaming algorithm for* TD *(and hence for* TC*) uses space $\Omega\left(\frac{m\sqrt{\Delta_V}}{T}\right)$. This proof uses a reduction of* BHM *to* TD.
4. *Any single-pass streaming algorithm for* TD *(and hence for* TC*) requires space $\Omega\left(\frac{m\Delta_E}{T} + \frac{m\sqrt{\Delta_V}}{T}\right)$. This follows from Parts 2 and 3. Note that we now have matching bounds for one-pass streaming algorithms for* TC.

7.2 Quantum Streaming for Triangle Counting and Distinguishing

Quantum streaming algorithms were first defined by Khadiev et al. [KKM18] (see also Ablayev et al. [AAKV18]). We will discuss modifying the proofs of the lower bounds for streaming on TD and TC from Theorem 10 to obtain lower bounds for quantum streaming for these problems.

Theorem 10.2 used that INDEX has communication complexity $\Omega(n)$. Fortunately, Ambainis et al. [ANTV02] showed that INDEX also has quantum communication complexity $\Omega(n)$. Hence we have the following analog to Theorem 10.2 by the same proof:

Theorem 11. *Any single-pass quantum streaming algorithm for* TD *(and hence for* TC*) requires space* $\Omega\left(\frac{m\Delta_E}{T}\right)$*. This proof uses a reduction of* INDEX *to* TD*. This follows from Theorem 10.2 and the work of Ambainis et al. [ANTV02].*

Can we do the same for Theorem 10.3? No. Gavinsky et al. [GKK$^+$08] showed that the quantum communication complexity of BHM is $O(\log n)$. Hence we do not have a non-trivial lower bound for TC or TD in the region where $\Delta_E = O(1)$ and $T = \Omega(n)$. Indeed, there is a quantum streaming algorithm that works well in that region. Kallaugher [Kal21] showed the following.

Theorem 12. *Restrict* TC *to the graphs where* $\Delta_E = O(1)$*,* $\Delta_V = \Omega(T)$*, and* $T = \Omega(m)$*. There is a single-pass quantum streaming algorithm for* TC *that uses space* $\tilde{O}(m^{2/5})$*.*

Open 2. *Find a lower bound of the form* $\Omega(m^c)$ *for* TC *in the case where* $\Delta_E = O(1)$*,* $\Delta_V = \Omega(T)$*, and* $T = \Omega(m)$*.*

7.3 Classical Streaming for k-Clique Counting and Distinguishing

In this section, we define two problems for k-clique finding which are analogous to Triangle Counting and Triangle Distinguishing.

Problem 7.4. k-Clique Counting *(*kCC*)*
INSTANCE: Graph $G = (V, E)$ *and* $k \in \mathbb{N}$*.*
QUESTION: Approximate the number of cliques of size k *in* G*.*

Problem 7.5. k-Clique Distinguishing *(*kCD*)*
INSTANCE: Graph $G = (V, E)$*,* $C \in \mathbb{N}$*, and the promise that* G *has either 0* k*-cliques or* $\geq C$ k*-cliques.*
QUESTION: Determine if G *has 0* k*-cliques or* $\geq C$ k*-cliques.*

Clearly kCD $\leq$ kCC. Hence a lower bound on kCD implies a lower bound on kCC.

Theorem 10.2 stated a $\Omega\left(\frac{m\Delta_E}{T}\right)$ space lower bound for single-pass streaming algorithms for Triangle Distinguishing. A similar proof gives the same lower bound for k-Clique Distinguishing (with T being the number of k-cliques); however this gives a trivial lower bound on most graphs, since Δ_E is usually small. We want a stronger lower bound for more general graphs. Additionally, since the quantum streaming complexity of triangle counting in the parameter setting $\Delta_E = O(1)$ and $T = \Omega(m)$ is an open problem it might be instructive to look for lower bounds on k-clique counting for $k \geq 4$ in this parameter setting to understand if the difficulty of this problem is unique for triangle counting.

For the next exercise you need the following definition and theorem.

Definition 3. *Let* $k, n \in \mathbb{N}$*.*

1. *A* ***perfect hypermatching M over $[kn]$*** *is a set of n ordered k-tuples $(i_1, \ldots, i_k)$, where $i_1, \ldots, i_k$ are distinct elements of $[kn]$, such that every $\ell \in [kn]$ is in exactly 1 ordered k-tuple.*

2. *Let M be a perfect hypermatching M over $[kn]$. We identify M with the following $n \times kn$ matrix: For every ordered k-tuple $(i_1, \ldots, i_k)$ in the hypermatching there is a row with 1's in the i_1th, i_2th, ..., i_kth spot, and 0's everywhere else. Note that a perfect hypermatching can be associated to many different matrices. We will turn this around: we will give Bob a perfect hypermatching by giving him a matrix.*

Problem 7.6. BOOLEAN HIDDEN HYPERMATCHING BHHM

INSTANCE: Alice gets a string $x \in \{0,1\}^{kn}$. Bob gets (a) a perfect hypermatching M over $[kn]$ via a matrix as described in Definition 3, and (b) a string $w \in \{0,1\}^n$ where w is promised to satisfy either $Mx = w$ or $Mx = \overline{w}$ (where $\overline{w}$ is w with every bit flipped).

QUESTION: Determine which is the case: $Mx = w$ or $Mx = \overline{w}$.

Theorem 13.

1. *(Verbin & Yu [VY11]) The randomized one-way communication complexity for* BHHM*, with Alice sending, is $\Omega(n^{1-(1/k)})$.*

2. *(Shi et al. [SWY15]) the quantum one-way communication complexity for* BHHM*, with Alice sending, is $\Omega(n^{1-(2/k)})$.*

Exercise 1.

1. *Prove that any classical single-pass streaming algorithm for* KCD *requires space $\Omega\left(m^{1-1/k}\right)$. (Hint: Use the lower bound on* BHHM *from Theorem 13.1).*

2. *Prove that any quantum single-pass streaming algorithm for* KCD *requires space $\Omega\left(m^{1-2/k}\right)$ (space is measured in qubits). (Hint: Use the lower bound on* BHHM *from Theorem 13.2).*

Open 3.

1. *We have looked at counting and detecting triangles and k-cliques. Look at the problems of counting and detecting other subgraphs such as k-cycles.*

2. *Obtain classical and quantum upper and lower bounds on p-pass streaming algorithms.*

7.4 Separations

So far we have not presented a large separations between classical and quantum streaming algorithms. Note that we have been looking at *natural* streaming problems. There are results for contrived problems.

Theorem 14.

1. *(Le Gall [Gal09]) There exists a streaming problem which (a) any classical algorithm requires $\Omega(n^{1/3})$ space, (b) there is a quantum algorithm that uses $O(\log n)$ space.*

2. *(Gavinsky et al. [GKK+08]) There exists a streaming problem which (a) any classical algorithm requires $\Omega(n^{1/2})$ space, (b) there is a quantum algorithm that uses $O(\log n)$ space.*

(There are reasons why the first result is not quite comparable to the second result.)

Open 4. *Find natural streaming problems for which there is a large separation between classical and quantum algorithms.*

Upshot Lower bounds on classical or quantum streaming algorithms are obtained by lower bounds on classical or quantum communication complexity. Hence the difficulty in obtaining a separation for streaming algorithms is to find a separation for communication complexity problems. This has been done for some contrived streaming problems; however, we would like to have a separation for natural problems.

8 MIP* = RE

Lets consider 3COL $\in$ **NP** as a game involving two people: an all powerful prover and a poly time verifier. The prover wants to convince the verifier that a given graph is 3-colorable.

- The prover sends the verifier a string y that he hopes will convince the verifier that $x \in A$. The obvious thing to send is a 3-coloring of G.
- The verifier then determines if y really is a 3-coloring of G. If so then he accepts that G is 3-colorable. If not then he now believes G is not 3-colorable.

Note that (a) there is only one prover, (b) the conversation is only one direction (prover sends a string to verifier), (c) the verifier can implement any deterministic polynomial time, and (d) the verifier is convinced $G \in$ 3COL iff $G \in$ 3COL.

We can modify the game by (1) allowing more rounds, (2)allowing the verifier to flip coins, (3) allowing the verifier a small probability of error. Such games are called interactive proof systems Note that an interactive proof system has one prover and one verifier. A multiprover interactive proof systems allows many provers, who cannot talk to each other.

Multiprover interactive proof systems have been used to get some lower bounds on how well a problem can be approximated in poly time, and can be considered a precursor to PCP (which you saw in Chapter **??** and the unique games conjecture (which you say in Chapter **??**.

Definition 4.

1. *A set A is in **MIP** if there is a Multiprover interactive system such that (a) if $x \in A$ then the verifier accepts, and (b) if $x \notin A$ then the verifier rejects with probability ≥ 0.9.*
2. *If we allow the provers to share entangled quantum states (the verifier is still classical) then this is **MIP***.*

MIP and MIP* differ a lot:

Theorem 15.

1. *(Babai et al. [BFL91]) MIP = NEXP.*
2. *(Ji et al. [JNV$^+$21]) MIP* = RE where RE is the first level of the arithmetic hierarchy, Since RE contains the Halting set, MIP* contains sets that are undecidable.*

Upshot MIP and MIP* give an example where in a classical setting problems are in NEXP and in quantum setting, problems can be undecidable.

9 Quantum Games

In the previous sections we measured how well an algorithm did by how much time or space it used (queries can be considered time). In this section we look at games and measure how well the players do by looking at their probability of winning.

We discuss two games such that if the players play the game with quantum resources they can provably do better than if they play the game with classical resources. For further discussion of these games, and other games with this property, see the survey of Brunner et al. [BCP+14].

9.1 The CHSH Game

Clauser, Horne, Shimony, and Holt [CHSH69] invented the CHSH GAME as a realizable experiment that can differentiate quantum from classical computing. (They did not give it that name; however, named using their initials.) We note that Clauser won the 2022 Nobel prize in Physics for this and other work [Rel22].

Problem 9.1. *The* CHSH GAME

INSTANCE: Alice gets bit x, Bob gets bit y. Before they get their bits they can discuss strategy.
QUESTION: Alice outputs bit a, Bob outputs bit b. Alice and Bob win iff $x \wedge y = a \oplus b$.

Clauser et al. [CHSH69] proved the following (see also Aaronson [Aar16, Chapter 13] for an exposition).

Theorem 16.

1. *If Alice and Bob play the* CHSH GAME *with classical resources (a) there is a deterministic strategy where they win with probability* 0.75 *(both always output 0), (b) there is no strategy, deterministic or randomized, that does better.*

2. *If Alice and Bob play the* CHSH GAME *with quantum resources (they prepare entangled qubits before the game begins) then (a) there is a strategy where they win with probability* $0.5+\sqrt{2}/4 \approx 0.85$ *(this is complicated), (b) there is no strategy that does better.*

This game is of interest since it is a case where the quantum world is provably different from the classical world. Note that the gap between the classical and quantum is $0.85 - .0.75 = 0.1$.

9.2 Magic Square Game

Cabello [Cab01] defined the Magic Square Game, though he did not call it that. For more information on it also see the survey of Brassard et al. on Quantum pseudo-telepathy [BBT05, Section 5] or the Wikipedia entry on Quantum pseudo-telepathy [Wik].

Problem 9.2. *The* MAGIC SQUARE GAME *(*MS GAME*)*

INSTANCE: Alice gets $i \in \{1,2,3\}$, Bob gets $j \in \{1,2,3\}$. They interpret i as the row of a 3×3 matrix and j as the column of a 3×3 matrix. Alice and Bob get to discuss strategy ahead of time.

QUESTION: Alice and Bob both output a three-bit sequence. Alice's sequence is used as the ith row of a matrix. Bob's sequence is used as the jth column of a matrix. If the following three conditions hold then Alice and Bob win, else they lose. (a) The values in row i add to an even number, (b) The values in column j add to an odd number. (c) Alice and Bob's values are consistent (they agree at (i,j)).

Theorem 17.

1. *If Alice and Bob play the* MS Game *with classical resources (a) there is a deterministic strategy where they win with probability* $\frac{8}{9} = 0.88\ldots$, *(b) there is no strategy, deterministic or randomized, that does better.*
2. *If Alice and Bob play the* MS Game *with quantum resources (they prepare entangled qubits before the game begins) then there is a strategy that wins with probability 1 (so always wins).*

This game is of interest since it is a case where the quantum world is provably different from the classical world. Note that the gap between the classical and quantum is $1.0 - 0.88\ldots = 0.11\ldots$.

Is there an interesting version of the MS Game on $k \times k$ matrices for $k \geq 4$? The following exercise shows that the answer is no.

Exercise 2.

1. *Give a* 4×4 *matrix* M *of 0's and 1's such that every row sums to an even number and every column sums to an odd number.*
2. *Show that there is a classical strategy for the* 4×4 MS Game *that wins with probability 1. (Hint: Use Part 1.)*
3. *Show that, for all* $k \geq 4$, *there is a classical strategy for the* $k \times k$ MS Game *that wins with probability 1.*

9.3 Comparing the CHSH Game with The MS Game

We give two reasons why the MS Game game is better for distinguishing classical and quantum computation, and one reason why the CHSH Game game is better.
Two reasons why the MS Game is better

1. In the CHSH Game the gap between the classical and quantum players is 0.1. In the MS Game the gap between the classical and quantum players is 0.11 which is bigger!
2. For the MS Game the quantum players *always* win. This is better for repeated experiments. Assume the game is played n times.

   (a) For the MS Game:
   If Alice and Bob are classical then the expected number of wins is $8n/9$.
   If Alice and Bob are quantum then the expected number of wins is n.
   So if Alice and Bob lose just once, then they must be classical.

   (b) For the CHSH Game
   If Alice and Bob are classical then the expected number of wins is $0.75n$.
   If Alice and Bob are classical then the expected number of wins is $0.85n$.
   These two cases are harder to distinguish since a lose by Alice and Bob could happen in either case.

One reason why the CHSH Game is better. Quantum computers (in 2023) are noisy. The computations are not that reliable. Hence many trials must be run. There is a lot of work on quantum error correction to try to alleviate this.

The CHSH GAME game is simpler and uses fewer operations, hence less noise. This is not just theoretical. The CHSH GAME is currently used in quantum systems in order to calibrate the quantum computers. The calculations done above for the MS GAME were assuming an error-free quantum computer which is not a reality yet.

In 2023 CHSH GAME is better and, as noted above, is actually used. However, if quantum hardware and out level of control on it improve, there may be a time in the future where the MS GAME is better.

Upshot There are games where if the players can use quantum entanglement then their probability of winning is provably higher than if they cannot.

10 Quantum MAXCUT

The title of this section might confuse people into thinking that there is a quantum algorithm for MAXCUT which is **NP**-complete. This is not the case. Instead, in this section we look at a quantum version of MAXCUT. This section differs from the previous ones in that rather than taking and seeing how well it can be solved with a quantum algorithm, we are taking a quantum problem, QMAXCUT, and seeing how well it can be solved with quantum techniques. We will first recall MAXCUT.

10.1 Classical MAXCUT

We state a variant of the classical MAXCUT problem. We will later see that the known upper and lower bounds for the MAXCUT problem we stated in Chapter **??** holds for this version.

Definition 5. *Let $G = (V, E, w)$ be a weighted graph where all the weights are ≥ 0 and sum to 1. We will view the weights as a probability distribution on the edges. A* ***cut*** *is a function, $f : V \to \{1, -1\}$. The* ***value*** *of a cut is given by*

$$\mathbb{E}\left[\frac{1}{2} - \frac{1}{2}f(u)f(v)\right] \tag{1}$$

where the expected value is over the edges of G via the distribution. Note that a cut can be viewed as assigning to each vertex a bit, even though the bits are in $\{-1, 1\}$ rather than the traditional $\{0, 1\}$.

Problem 10.1. MAXCUT
INSTANCE: $G = (V, E, w)$ a weighted graph where all the weights are ≥ 0 and sum to 1.
QUESTION: Find the value of the largest cut.

Exercise 3. *Show that the problem* MAXCUT *defined in this section is equivalent to the* MAXCUT *problem defined in Chapter* **??**.

By Exercise 3 all of the known lower bounds stated for MAXCUT earlier in this book hold here. This yields the first two items in the next theorem.

Theorem 18.

1. *(Hastad [Has01, Theorem 8.2] and Trevisan et al. [TSSW00, Theorem 4.4]) Assume* $\mathbf{P} \neq \mathbf{NP}$*. There does not exist an $\epsilon > 0$, and a polynomial time algorithm, that returns $\geq (\frac{16}{17} + \epsilon)\text{OPT}$. Note that $\frac{16}{17} \approx 0.9411$.*

2. *(Khot et al. [KKMO07], O'Donnell & Wu [OW08], Khot & O'Donnell [KO09]) Assume the Unique Games Conjecture holds. There does not exist an $\epsilon > 0$, and a polynomial time algorithm, that returns $\geq (0.87856\ldots + \epsilon)\mathrm{OPT}$. See Chapter* **??** *for the exact constant.*

3. *(Goemans & Williamson [GW95]) There is an algorithm that matches the lower bound in Part 2. The algorithm given in Chapter* **??** *for* MAXCUT *can easily be modified for the version given in this section. Recall that the algorithm used a Semi Definite Program (*SDP*).*

In summary, the SDP approach to (classical) MAXCUT is optimal assuming the Unique Games Conjecture.

10.2 Quantum MAXCUT (qMAXCUT)

There is a quantum version of MAXCUT which we call qMAXCUT. We will not define it. We will state theorems that and contrast it to the classical MAXCUT. For more background see either Carolan & Dontha [CD22] or the references in Theorem 19.

Problem 10.2. qMAXCUT
INSTANCE: $G = (V, E, w)$ a weighted graph where all the weights are ≥ 0 and sum to 1.
QUESTION: Find the value of the largest quantum cut.

Theorem 19. *The problem we are considering is* qMAXCUT*.*

1. *(Briët et al. [BdOFV10], last page where they state $u(3)$) There is a polynomial time algorithm, that returns $\geq (0.956\ldots)\mathrm{OPT}$ for the restricted version where we only seek Product State Solutions. This algorithm uses* SDP *techniques. (See the paper for the exact constant.)*

2. *(Hwang et al. [HNP+21]) Assume the Unique Games Conjecture holds. The result in Part 1 is optimal for the restricted version where we only seek Product State Solutions.*

3. *(Gharibian & Parekh [GP19]) There is a polynomial time algorithm, that returns $\geq (0.498\ldots)\mathrm{OPT}$. This algorithm uses* SDP *techniques.*

4. *(Hwang et al. [HNP+21]) If you use standard techniques for* SDP *based algorithm then, assuming the Unique Games Conjecture and a plausible inequality (a generalization of Borell's inequality to vectors—see Hwang [HNP+21, Conjecture 1.1]), you can do no better than the algorithm in Part 3.*

5. *(Anshu et al. [AGM20]) There is a polynomial time algorithm, that returns $\geq (0.53\ldots)\mathrm{OPT}$.*

Parts 4 and 5 are interesting because they give the following contrast:

1. For approximating MAXCUT in polynomial time, SDP techniques are optimal (assuming the Unique Games Conjecture).

2. For approximating qMAXCUT in quantum polynomial time, SDP techniques are not optimal.

We give a potential contrast:

1. Feige et al. [FKL02] showed that if MAXCUT is restricted to graphs of bounded degree then there are approximations better than that in $0.87856\mathrm{OPT}$, which beats the lower bound for the general case stated in Theorem 18. So MAXCUT is easier if the graphs are of low degree.

2. Brandao and Harrow [BH18] proved lower bounds on approximating qMAXCUT for graph of low degree. So qMAXCUT seems harder if the graphs are of low degree.

We close this section with the obvious open problem:

Open 5. *Obtain closer upper and lower bounds for how well* qMAXCUT *can be approximated in polynomial time.*

Upshot The problems MAXCUT and qMAXCUT differ in several ways: (a) techniques needed for approximating in poly time (assuming the Unique Games Conjecture), and (b) MAXCUT seems easier on low degree graphs, whereas qMAXCUT seems harder. These observations indicate differences between the classical world and the quantum world.

11 Quantum Supremacy

The original point of quantum computing is that it should be better than classical computing on some problems. This goal was crystallized by John Preskill [Pre12] (see also [Pre19]) who defined ***Quantum Supremacy***.

Definition 6. ***Quantum Supremacy*** *is the goal of finding a problem where a quantum computer can outperform a classical computer. Note that this involves (a) formulating a problem, (b) come up with a quantum algorithm for it, (c) code up that algorithm on a real quantum computer, (d) have an argument (it need not be a proof) that any classical algorithm, when coded up, will do much worse than the quantum algorithm. Note that this is not theoretical. The goal involves real quantum computers versus real classical computers. The goal is that the quantum computer is* ***better*** *than the classical one; however,* ***better*** *may be speed or energy or some other parameter.*

In this chapter we discussed many problems where quantum algorithms seem to do better than any known classical algorithm. However, putting these quantum algorithms on a real quantum computer is difficult. Hence other problems have been suggested as candidates for achieving quantum supremacy. We discuss two problems for which quantum supremacy may have been achieved.

11.1 Quantum Sampling

Problem 11.1. Random Circuit Sampling *(*RCS*)*

INSTANCE: A quantum circuit C. C takes n qubits as input and outputs n qubits that are then measured to get an element of $\{0,1\}^n$. Note that if you run C on the same input twice you may not get the same answer. Hence $C(0^n)$ can be viewed as a distribution rather than an element of $\{0,1\}^n$. We call this distribution D. If the quantum circuit is chosen at random then the D is far from uniform.

QUESTION: (Informally) Take enough samples of the circuit to show statistically that D is unlikely to be uniform.

The following are known.

1. Bouland et al. [BFNV18] and Aaronson & Chen [AC17] give evidence that if RCS is classically easy then certain reasonable complexity hypothesis are false. Hence RCS is probably hard classically. We note that this is a theoretical asymptotic result so it may not be as helpful for quantum supremacy as it appears at first glance.

2. On October 23, 2019 Google published a paper [AA19] which claims to have have solved the 53-bit RCS problem in 200 seconds. They also claim that any classical supercomputer would have taken 10,000 years. The quantum processor used is named ***Sycamore***.

3. IBM [PGA19] (see also Scott Aaronson's blog post [Aar19]) showed how the computation would have taken only 2.5 days on a supercomputer.

4. We will stop here; however, the claims and counter-claims about if Google really did achieve quantum supremacy continue. For intelligent discussions of the issue go to Scott Aaronson's Blog and search for `Quantum Supremacy`.

11.2 Gaussian Boson Sampling

Aaronson & Arkhipov [AA13] proposed BOSON SAMPLING as a candidate problem for quantum supremacy. We omit the description as it is somewhat technical, They give evidence that if BOSON SAMPLING is easy in one way then the polynomial hierarchy collapses, and if it is easy in another way then (with some assumptions) computing the permanent of a matrix is easy, which also implies that the polynomial hierarchy collapses.

Hamilton et al. [HKS+17] introduced a related problem, GAUSSIAN BOSON SAMPLING. They give evidence that if GAUSSIAN BOSON SAMPLING is easy then computing the permanent of a matrix is easy, so the polynomial hierarchy collapses.

The following are known.

1. In December of 2020, a group based in the University of Science and Technology of China (USTC) [Poa20] achieved quantum supremacy by implementing GAUSSIAN BOSON SAMPLING on 76 photons with their photonic quantum computer Jiuzhang. The paper states that to generate the number of samples the quantum computer generates in 2 seconds, a classical supercomputer would require 600 million years of computation. For more information on these results see the articles by Conover [Con21] and Garisto [Gar20].

2. They later increased the number of photons to 113.

3. Martinez-Cifuentes et al. [MCFRQ22] give reasons why the USTC group may not have actually achieved quantum supremacy.

11.3 Has Quantum Supremacy Happened?

We close with a quote from a blog post of Scott Aaronson [Aar22]. We insert comments in parenthesis for clarity.

> The experiments by Google and USTC and now Xanadu (they worked on a graph problem that has a classical $O(n^3)$ algorithm so the quantum advantage cannot be too large) represent a big step forward for the field, but since they started being done, the classical spoofing attacks have also steadily improved, to the point that whether "quantum computational supremacy" still exists depends on exactly how you define it.
>
> Briefly: If you measure by total operations, energy use, or CO2 footprint, then probably yes, quantum supremacy remains. But if you measure by number of seconds, then it doesn't remain, not if you're willing to shell out for enough cores on AWS (Amazon Web Services) or your favorite supercomputer. And even the quantum supremacy that does remain might eventually fall to, e.g., further improvements due to Gao et al. [GKC+21].

> For more details, see e.g., the now-published work of Pan, Chen and Zhang [PCZ22] or this good popular summary by Adrian Cho [Cho22] for Science.

Scott later argues that what is really needed are new quantum supremacy experiments.

References

[AA13] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. *Theory Comput.*, 9:143–252, 2013. https://arxiv.org/pdf/1011.3245.pdf. doi:10.4086/toc.2013.v009a004.

[AA19] Frank Arute and 77-Authors. Quantum supremacy ausing a programmable supercomuter processor. *Nature*, 574:505–501, 2019. doi:10.1038/s41586-019-1666-5.

[AAKV18] Farid M. Ablayev, Marat Ablayev, Kamil Khadiev, and Alexander Vasiliev. Classical and quantum computations with restricted memory. In Hans-Joachim Böckenhauer, Dennis Komm, and Walter Unger, editors, *Adventures Between Lower Bounds and Higher Altitudes - Essays Dedicated to Juraj Hromkovič on the Occasion of His 60th Birthday*, volume 11011 of *Lecture Notes in Computer Science*, pages 129–155. Springer, 2018. URL: https://doi.org/10.1007/978-3-319-98355-4_9, doi:10.1007/978-3-319-98355-4_9.

[Aar13] Scott Aaronson. *Quantum Computing since Democritus*. Cambridge University Press, 2013.

[Aar16] Scott Aaronson. Introduction to quantum information science, 2016. https://www.scottaaronson.com/qclec.pdf.

[Aar19] Scott Aaronson. Quantum supremacy: the glove are off, 2019. https://scottaaronson.blog/?p=4372.

[Aar22] Scott Aaronson. Summer 2022 quantum supremacy updates, 2022. https://scottaaronson.blog/?p=6645.

[AC17] Scott Aaronson and Lijie Chen. Complexity-theoretic foundations of quantum supremacy experiments. In Ryan O'Donnell, editor, *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, volume 79 of *LIPIcs*, pages 22:1–22:67. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.CCC.2017.22.

[Adl94] Leonard M. Adleman. The function field sieve. In Leonard M. Adleman and Ming-Deh A. Huang, editors, *Algorithmic Number Theory, First International Symposium, ANTS-I, Ithaca, NY, USA, May 6-9, 1994, Proceedings*, volume 877 of *Lecture Notes in Computer Science*, pages 108–121. Springer, 1994. doi:10.1007/3-540-58691-1_48.

[AGM20] Anurag Anshu, David Gosset, and Karen Morenz. Beyond product state approximations for a quantum analogue of MAXCUT. In Steven T. Flammia, editor, *15th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2020, June 9-12, 2020, Riga, Latvia*, volume 158 of *LIPIcs*, pages 7:1–7:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.TQC.2020.7.

[AH99] Leonard M. Adleman and Ming-Deh A. Huang. Function field sieve method for discrete logarithms over finite fields. *Information and Computation*, 151(1-2):5–16, 1999. doi:10.1006/inco.1998.2761.

[AHWW16] Amir Abboud, Thomas Dueholm Hansen, Virginia Vassilevska Williams, and Ryan Williams. Simulating branching programs with edit distance and friends: or: a polylog shaved is a lower bound made. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 375–388. ACM, 2016. https://arxiv.org/abs/1511.06022. doi:10.1145/2897518.2897653.

[AN20] Alexandr Andoni and Negev Shekel Nosatzki. Edit distance in near-linear time: it's a constant factor. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 990–1001. IEEE, 2020.
https://arxiv.org/abs/1109.5635. doi:10.1109/FOCS46700.2020.00096.

[ANTV02] Andris Ambainis, Ashwin Nayak, Amnon Ta-Shma, and Umesh V. Vazirani. Dense quantum coding and quantum finite automata. *Journal of the ACM*, 49(4):496–511, 2002. doi:10.1145/581771.581773.

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh V. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, 1997.
https://arxiv.org/abs/quant-ph/9701001. doi:10.1137/S0097539796300933.

[BBT05] Gilles Brassard, Anne Broadbent, and Alain Tapp. Quantum pseudo-telepathy. *Foundations of Physics (Special Asher Peres Memorial Issue)*, 35(11):1877–1907, 2005.
https://arxiv.org/abs/quant-ph/0407221. doi:10.1007/s10701-005-7353-4.

[BCDP96] David Beckman, Amalavoyal Chari, Srikrisha Devabhaktuni, and John Preskill. Efficient networks for quantum factoring. *Physical Review Letters*, 54(2):1034–1063, 1996.
https://arxiv.org/abs/quant-ph/9602016. doi:10.1103/PhysRevA.54.1034.

[BCP+14] Nicolas Brunner, Daniel Cavalcanti, Stefano Pironio, Valerio Scarani, and Stephanie Wehner. Bell nonlocality. *Review of Modern Physics*, 2014.
https://arxiv.org/abs/1303.2849. doi:10.1103/RevModPhys.86.419.

[BdOFV10] Jop Briët, Fernando Mário de Oliveira Filho, and Frank Vallentin. The positive semidefinite Grothendieck problem with rank constraint. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis, editors, *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part I*, volume 6198 of *Lecture Notes in Computer Science*, pages 31–42. Springer, 2010.
https://arxiv.org/abs/0910.5765. URL: https://doi.org/10.1007/978-3-642-14165-2_4.

[BEG+21] Mahdi Boroujeni, Soheil Ehsani, Mohammad Ghodsi, MohammadTaghi Hajiaghayi, and Saeed Seddighin. Approximating edit distance in truly subquadratic time: Quantum and mapreduce. *Journal of the ACM*, 68(3):19:1–19:41, 2021. doi:10.1145/3456807.

[BFL91] László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. *Compututational Complexity*, 1:3–40, 1991. doi:10.1007/BF01200056.

[BFNV18] Adam Bouland, Bill Fefferman, Chinmay Nirkhe, and Umesh Vazarani. On the complexity and verifiation of quantum random circuit sampling. *Nature Physics*, 15:159–163, 2018.
https://arxiv.org/abs/1803.04402. doi:10.1038/s41567-018-0318-2.

[BH18] Fernando Brandao and Aram Harrow. Product-state approximations to quantum ground states. *Communications in Mathematical Physics*, 342(1):47–80, 2018.
https://arxiv.org/abs/1310.0017. URL: https://dl.acm.org/doi/10.1145/2488608.2488719, doi:10.1145/2488608.2488719.

[BI15] Arturs Backurs and Piotr Indyk. Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 51–58. ACM, 2015. URL: https://doi.org/10.1145/2746539.2746612.

[BOV13] Vladimir Braverman, Rafail Ostrovsky, and Dan Vilenchik. How hard is counting triangles in the streaming model? In Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska, and David Peleg, editors, *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, volume 7965 of *Lecture Notes in Computer Science*, pages 244–254. Springer, 2013. URL: https://doi.org/10.1007/978-3-642-39206-1_21.

[Cab01] Adan Cabello. Bell's theorem without inequalities and without probabilities for two observers. *Physical Review Letters*, 2001.
https://arxiv.org/abs/quant-ph/0008085. doi:10.1103/PhysRevLett.86.1911.

[CCD$^+$03] Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. Exponential algorithmic speedup by a quantum walk. In Lawrence L. Larmore and Michel X. Goemans, editors, *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*, pages 59–68. ACM, 2003.
https://arxiv.org/abs/quant-ph/0209131. doi:10.1145/780542.780552.

[CCG23] Andrew M. Childs, Matthew Coudron, and Amin Shiraz Gilani. Quantum algorithms and the power of forgetting. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 37:1–37:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. URL: http://doi.10.4230/LIPIcs.ITCS.2023.37.

[CD22] Joseph Carolan and Suchetan Dontha. Hardness of approximation of quantum MAX-CUT, 2022.
https://www.cs.umd.edu/~gasarch/BLOGPAPERS/qmaxcut.pdf.

[CDG+20] Diptarka Chakraborty, Debarati Das, Elazar Goldenberg, Michal Koucký, and Michael E. Saks. Approximating edit distance within constant factor in truly subquadratic time. *J. ACM*, 67(6):36:1–36:22, 2020. doi:10.1145/3422823.

[CFG02] Andrew M. Childs, Edward Farhi, and Sam Gutmann. An example of the difference between quantum and classical random walks. *Quantum Inf. Process.*, 1(1-2):35–43, 2002. https://arxiv.org/abs/quant-ph/0103020. doi:10.1023/A\%3A1019609420309.

[Cho22] Adrian Cho. Ordinary computers can beat Google's quantum computers after all. *Science*, 377, 2022. URL: http://doi.10.1126/science.ade2364.

[CHSH69] John Clauser, Michael Horne, Abner Shimony, and Richard Holt. Proposed experiment to test local hidden-variable theories. *Physical Review Letters*, 23(15):880–884, 1969. doi:10.1103/PhysRevLett.23.880.

[Con21] Emily Conover. The new light-based quantum computer Jiuzhang has achieved quantum supremacy. *ScienceNews*, January 16, 2021, 2021.

[CvD10] Andrew Childs and Wim van Dam. Quantum algorithms for algebraic problems. *Review of Modern Physics*, 82:1–52, 2010.
https://arxiv.org/abs/0812.0380. doi:10.1103/RevModPhys.82.1.

[DdW11] Andrew Drucker and Ronald de Wolf. Quantum proofs for classical theorems. *Theory of Computing*, 2:1–54, 2011. doi:10.4086/toc.gs.2011.002.

[FKL02] Uriel Feige, Marek Karpinski, and Michael Langberg. Improved approximation of max-cut on graphs of bounded degree. *Journal of Algorithms*, 43(2):201–219, 2002.
https://www.sciencedirect.com/science/article/pii/S0196677402000056.
doi:10.1016/S0196-6774(02)00005-6.

[Gal09] François Le Gall. Exponential separation of quantum and classical online space complexity. *Theory Comput. Syst.*, 45(2):188–202, 2009.
https://arxiv.org/pdf/quant-ph/0606066.pdf. doi:10.1007/s00224-007-9097-3.

[Gar20] Daniel Garisto. Light-based quantum computer exceeds fastest classical supercomputers. *Scientific American*, December 3, 2020, 2020.

[Gas19] William I. Gasarch. Guest column: The third p=?np poll. *SIGACT News*, 50(1):38–59, 2019. https://www.cs.umd.edu/users/gasarch/papers/poll3.pdf. doi:10.1145/3319627.3319636.

[GKC+21] Xun Gao, Marcin Kalinowski, Chi-Ning Chou, Mikhail Lukin, Boaz Barak, and Soonwon Choi. Limitations of linear cross-entropy as a measure for quantum advantage, 2021.
https://arxiv.org/abs/2112.01657.

[GKK+08] Dmitry Gavinsky, Julia Kempe, Iordanis Kerenidis, Ran Raz, and Ronald de Wolf. Exponential separation for one-way quantum communication complexity, with applications to cryptography. *SIAM Journal of Computing*, 38(5):1695–1708, 2008.
https://doi.org/10.1137/070706550. doi:10.1137/070706550.

[GP19] Sevag Gharibian and Ojas Parekh. Almost optimal classical approximation algorithms for a quantum generalization of max-cut. In Dimitris Achlioptas and László A. Végh, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2019, September 20-22, 2019, Massachusetts Institute of Technology, Cambridge, MA, USA*, volume 145 of *LIPIcs*, pages 31:1–31:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. https://doi.org/10.4230/LIPIcs.APPROX-RANDOM.2019.31. doi:10.4230/LIPIcs.APPROX-RANDOM.2019.31.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 212–219. ACM, 1996. https://doi.org/10.1145/237814.237866. doi:10.1145/237814.237866.

[GW95] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995. URL: https://dl.acm.org/doi/pdf/10.1145/227683.227684, doi:10.1145/227683.227684.

[Has01] Johan Hastad. Some optimal inapproximability results. *Journal of the Association of Computing Machinery (JACM)*, 48(4):798–859, 2001. URL: https://dl.acm.org/doi/10.1145/502090.502098, doi:10.1145/502090.502098.

[HKS+17] Craig Hamilton, Regine Kruse, Linda Sansoni, Sonja Barkhofen, Christine Silverhorn, and Igor Jex. Gaussian boson sampling. *Physical Review Letters*, 119, 2017. https://arxiv.org/pdf/1612.01199.pdf. doi:10.1103/PhysRevLett.119.170501.

[HNP+21] Yeongwoo Hwang, Joe Neeman, Ojas Parekh, Kevin Thompson, and John Wright. Unique games hardness of quantum max-cut, and a vector-valued Borell's inequality, 2021. https://arxiv.org/abs/2111.01254.

[JK21] Rajesh Jayaram and John Kallaugher. An optimal algorithm for triangle counting in the stream. In Mary Wootters and Laura Sanità, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021, August 16-18, 2021, University of Washington, Seattle, Washington, USA (Virtual Conference)*, volume 207 of *LIPIcs*, pages 11:1–11:11. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.APPROX/RANDOM.2021.11.

[JNV+21] Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. Mip* = RE. *Commun. ACM*, 64(11):131–138, 2021. https://arxiv.org/abs/2001.04383. doi:10.1145/3485628.

[Jor11] Stephen Jordan. Quantum algorithms zoo, 2011. https://quantumalgorithmzoo.org/.

[JZ22] Stacey Jeffery and Sebastian Zur. Multidimensional quantum walks, with applications to k-distinctness, 2022. https://arxiv.org/abs/2208.13492.

[Kal21] John Kallaugher. A quantum advantage for a natural streaming problem. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2021*, pages 897–908. IEEE, 2021. doi:10.1109/FOCS52979.2021.00091.

[KKM18] Kamil Khadiev, Aliya Khadieva, and Ilanz Mannapo. Quantum online algorithms with respect to space and advice complexity. *Lobachevskii Journal of Mathematics*, 39(9):1377–1387, 2018. doi:10.1134/S1995080218090421.

[KKMO07] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM Journal on Computing*, 37(1):319–357, 2007. https://www.cs.cmu.edu/~odonnell/papers/maxcut.pdf. doi:10.1137/S0097539705447372.

[KO09] Subhash Khot and Ryan O'Donnell. SDP gaps and UGC-hardness for max-cut-gain. *Theory Comput.*, 5(1):83–117, 2009. https://doi.org/10.4086/toc.2009.v005a004. doi:10.4086/toc.2009.v005a004.

[KP17a] John Kallaugher and Eric Price. A hybrid sampling scheme for triangle counting. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1778–1797. SIAM, 2017. https://doi.org/10.1137/1.9781611974782.116. doi:10.1137/1.9781611974782.116.

[KP17b] Iordanis Kerenidis and Anupam Prakash. Quantum recommendation systems. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, volume 67 of *LIPIcs*, pages 49:1–49:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ITCS.2017.49.

[LL93] Arjen Lenstra and Hendrik Lenstra, editors. *The development of the number field sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer-Verlag, New York, Heidelberg, Berlin, 1993.

[MCFRQ22] Javier Martinez-Chfuentes, K.M. Fonseca-Romero, and Nicolas Quesada. Classical models are a better explanation of the Jiuzhang 1.0 Gaussian boson sampler, 2022. https://arxiv.org/pdf/2207.10058.pdf.

[Mer07] N. David Mermin, editor. *Quantum computer science: an introduction*. Cambridge Press, 2007.

[MLLL$^+$12] Enrique Martin-Lopez, Anthony Laing, Thomas Lawson, Roberto Alvarez, Xiao-Oi Zhou, and Jeremy O'Brian. Experimental realization of Shor's quantum factoring algorithm using qubit recycling. *Nature Photonics*, 6, 2012. https://arxiv.org/abs/1111.4147. doi:10.1038/414883a.

[NC16] Michael A. Nielson and Isaac L. Chuang. *Quantum Computation and Quantum Information (10th Anniversary edition)*. Cambridge University Press, 2016.

[OW08] Ryan O'Donnell and Yi Wu. An optimal sdp algorithm for max-cut, and equally optimal long code tests. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 335–344. ACM, 2008. https://doi.org/10.1145/1374376.1374425. doi:10.1145/1374376.1374425.

[PCZ22] Feng Pan, Keyang Chen, and Pan Zhang. Solving the sampling problem of the sycamore quantum circuits. *Phys. Rev. Lett.*, 129:090502, Aug 2022. https://arxiv.org/abs/2111.03011. URL: https://link.aps.org/doi/10.1103/PhysRevLett.129.090502, doi:10.1103/PhysRevLett.129.090502.

[PGA19] Edwin Pednault, John Gunnels, and Giacomo Scott Aaronson. Levarging secondary stoarge to simulate deep 54-qubit Sycamore circuits, 2019. https://arxiv.org/abs/1910.09534.

[Poa20] Jian-Wei Pan and 18 other authors. Quantum computational advantage using photons. *Science*, 370, 2020. https://arxiv.org/abs/2012.01625. URL: https://www.science.org/doi/10.1126/science.abe8770, doi:10.1126/science.abe8770.

[Pom96] Carl Pomerance. A tale of two sieves. *Notices of the American Mathematical Society*, 43:1473–1485, 1996. https://www.ams.org/notices/199612/pomerance.pdf.

[Pre12] John Preskill. Quantum computing and the entanglement frontier, 2012. https://arxiv.org/pdf/1203.5813.pdf.

[Pre19] John Preskill. Why I called it quantum supremacy. *Quanta*, 2019. https://www.quantamagazine.org/john-preskill-explains-quantum-supremacy-20191002/.

[Rel22] Press Release. The Nobel prize in physics in 2022, 2022. https://www.nobelprize.org/prizes/physics/2022/summary/.

[Rub18] Aviad Rubinstein. Approximaing edit distance, 2018. https://theorydish.blog/2018/07/20/approximating-edit-distance/.

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 124–134. IEEE Computer Society, 1994. /10.1109/SFCS.1994.365700. URL: https://doi.org/10.1109/SFCS.1994.365700.

[Sho99] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.*, 41(2):303–332, 1999. doi:10.1137/S0036144598347011.

[SSV13] John Smolin, Graeme Smith, and Alexander Vargo. Oversimplifying quantum factoring. *Nature*, 499:163–165, 2013. doi:10.1038/nature12290.

[SWY15] Yaoyun Shi, Xiaodi Wu, and Wei Yu. Limits of quantum one-way communication by matrix hypercontractive inequality, 2015. https://www.cs.umd.edu/~xwu/papers/GHM_v4.pdf.

[Tan19] Ewin Tang. A quantum-inspired classical algorithm for recommendation systems. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 217–228. ACM, 2019. https://eccc.weizmann.ac.il/report/2018/128. URL: https://dl.acm.org/doi/10.1145/3313276.3316310, doi:10.1145/3313276.3316310.

[TSSW00] Luca Trevisan, Gregory B. Sorkin, Madhu Sudan, and David P. Williamson. Gadgets, approximation, and linear programming. *SIAM J. Comput.*, 29(6):2074–2097, 2000. doi:10.1137/S0097539797328847.

[VY11] Elad Verbin and Wei Yu. The streaming complexity of cycle counting, sorting by reversals, and other problems. In Dana Randall, editor, *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 11–25. SIAM, 2011. doi:10.1137/1.9781611973082.2.

[Wag13] Samuel Wagstaff. *The joy of factoring.* AMS, Providence, 2013.

[Wik] Wikipedia. Quantum psuedo-telepathy. https://en.wikipedia.org/wiki/Quantum_pseudo-telepathy#.