Diagonalization Games

Noga Alon **Olivier Bousquet** **Kasper Green Larsen**
Shay Moran **Shlomo Moran**

Abstract. We study several variants of a combinatorial game which is based on Cantor's diagonal argument. The game is between two players called Kronecker and Cantor. The names of the players are motivated by the known fact that Leopold Kronecker did not appreciate Georg Cantor's arguments about the infinite, and even referred to him as a 'scientific charlatan'.

In the game Kronecker maintains a list of m binary vectors, each of length n, and Cantor's goal is to produce a new binary vector which is different from each of Kronecker's vectors, or prove that no such vector exists. Cantor does not see Kronecker's vectors but he is allowed to ask queries of the form

"*What is bit number j of vector number i?*"

What is the minimal number of queries with which Cantor can achieve his goal? How much better can Cantor do if he is allowed to pick his queries *adaptively*, based on Kronecker's previous replies?

The case when $m = n$ is solved by diagonalization using n (non-adaptive) queries. We study this game more generally, and prove an optimal bound in the adaptive case and nearly tight upper and lower bounds in the non-adaptive case.

1. INTRODUCTION. The concept of infinity has been fascinating philosophers and scientists for hundreds, perhaps thousands of years. The work of Georg Cantor (1845 – 1918) played a pivotal role in the mathematical treatment of the infinite. Cantor's work is based on a natural idea which asserts that two (possibly infinite) sets have the same size whenever their elements can be paired in one-to-one correspondence with each other [**2**]. Despite its simplicity, this notion has counter-intuitive implications: for example, a set can have the same size as a proper subset of it[1]; this phenomenon is nicely illustrated by *Hilbert's paradox of the Grand Hotel*, see e.g., [**6**].

This simple notion led Cantor to develop his theory of sets, which forms the basis of modern mathematics. Alas, Cantor's set theory was controversial at the start, and only later became widely accepted:

*The objections to Cantor's work were occasionally fierce: Leopold Kronecker's public opposition and personal attacks included describing Cantor as a 'scientific charlatan', a 'renegade'and a 'corrupter of youth'. Kronecker objected to Cantor's proofs that the algebraic numbers are countable, and that the transcendental numbers are uncountable, results now included in a standard mathematics curriculum [**3**] .*

Cantor's work influenced areas outside pure mathematics. For example, Cantor's diagonal argument is employed in the theory of computation to prove that there are problems that cannot be solved by any algorithm. This idea underpins the concept of undecidable problems, a cornerstone in theoretical computer science. In complexity theory, sophisticated variants of the diagonal argument are used to establish separations

[1] E.g., the natural numbers and the even numbers, via the correspondence "$n \mapsto 2n$".

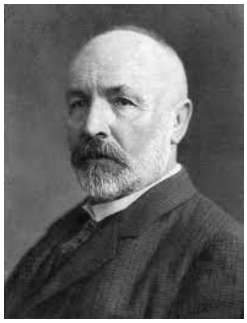

Figure 1. Georg Cantor (1845 – 1918).
Source: en.wikipedia.org (US-PD)

Figure 2. Leopold Kronecker (1823 – 1891).
Source: en.wikipedia.org (US-PD)

between a variety of complexity classes (see e.g., Chapter 5 of [**9**]). Interestingly, the combinatorial diagonalization game we investigate in this paper, which studies diagonalizations in *finite* sets, has applications in complexity theory. It was introduced in this context by Vyas and Williams [**8**] as a means for establishing lower bounds on circuit complexity. We will discuss this more comprehensively later in this section as well as in Section 5.

Diagonalization. One of the most basic and compelling results in set theory is that not all infinite sets have the same size. To prove this result, Cantor came up with a beautiful argument, called diagonalization. This argument is routinely taught in introductory classes to mathematics, and is typically presented as follows. Let $\mathbb{N} = \{1, 2, 3, \ldots\}$ denote the set of natural numbers and let $\{0,1\}^{\mathbb{N}}$ denote the set of all infinite binary vectors. Clearly both sets are infinite, but it turns out that they do not have the same size: assume towards a contradiction that there is a one-to-one correspondence $i \mapsto v_i$, where $v_i = (v_i(1), v_i(2), \ldots)$ is the infinite binary vector corresponding to $i \in \mathbb{N}$. Define a vector

$$u = (1 - v_1(1), 1 - v_2(2), \ldots).$$

That is, u is formed by letting its jth entry be equal to the negation of the jth entry of v_j.

Notice that this means that the resulting vector u disagrees with v_i on the ith entry, and hence $u \neq v_i$ for all i. Thus, we obtain a binary vector which does not correspond to any of the natural numbers via the assumed correspondence—a contradiction.

Rather than reaching a contradiction, it is instructive to take a positivist perspective according to which diagonalization can be seen as a constructive procedure that does the following:

> *Given binary vectors $v_1, v_2, \ldots$, find a binary vector u such that $u \neq v_i$ for all i.*

Moreover, notice that Cantor's diagonal argument involves querying only a single entry for each of the input vectors (i.e., the "diagonal" entries $v_j(j)$). Thus, it is possible to construct u while using only a little information about the input vectors v_i's (a single bit per vector).

In this manuscript we study a finite variant of the problem in which m binary vectors $v_1, \ldots, v_m$ of length n are given and the goal is to produce a vector u which is different from all of the v_i's, or to report that no such vector exists, while querying as few as

possible of the entries of the v_i's. We first study the case when $m < 2^n$ whence such a vector u is guaranteed to exist, and the goal boils down to finding one, and later the case when $m \geq 2^n$.

Cantor-Kronecker Game and Complexity Theory. As previously mentioned, finite variants of Cantor diagonalization, similar to the Cantor-Kronecker game, have been utilized in complexity theory (e.g., [**7, 8**]). In a recent work, Vyas and Williams explicitly define and analyze a variant of the Cantor-Kronecker game, which they call the *"Missing String"* problem [**8**]. They use this game as a unifying tool for establishing separations between complexity classes defined by circuits, and also for deriving novel ones. This unifying tool employs methods for minimizing the number of queries needed to determine a pre-designated entry of a specific missing vector.

How is this variant of the Cantor-Kronecker game used to derive separations in complexity theory? In a nutshell, we are given a family $\mathcal{F}$ of bounded computation devices (e.g., bounded size/depth circuits), and each of the vectors $v_1, \ldots, v_m$, for $m < 2^n$, corresponds to a subset X of $\{0,1\}^k$ which can be computed by a device in $\mathcal{F}$ (i.e., $n = 2^k$). So, a vector u that differs from all v_i vectors signifies a *hard* subset Y of $\{0,1\}^k$ that cannot be computed by a device in $\mathcal{F}$. Finally, a computation device which is capable of efficiently computing each bit of u by examining a handful of bits from the v_i vectors, computes the set Y with a relatively small complexity. We will go back to this variant of the game in the last section of this manuscript, where we suggest possible directions for future work.

2. THE CANTOR-KRONECKER GAME. Consider a game between two players called Kronecker and Cantor. In the game there are two parameters m and n, where m, n are positive integers. Kronecker maintains a set $V = \{v_1, v_2, \ldots, v_m\}$ of m binary vectors, each of length n. Cantor's goal is to produce a binary vector u, also of length n, which differs from each v_i, or to report that no such vector exists. To do so, he is allowed to ask queries, where each query is of the form

"What is bit number j of vector number i?",

where $1 \leq j \leq n$, $1 \leq i \leq m$. Kronecker answers each query posed by Cantor. The objective of Cantor is to minimize the number of queries enabling him to produce u, whereas Kronecker tries to maximize the number of queries. We distinguish between two versions of the game:

- In the *adaptive* version Cantor presents his queries to Kronecker in a sequential manner, and may decide on the next query as a function of Kronecker's answers to the previous ones.
- In the *oblivious* (i.e., non-adaptive) version Cantor must declare all of his queries in advance, before getting answers to any of them.

For $m \leq n$, the smallest number of queries, both in the adaptive and oblivious versions, is m. Indeed, Cantor can query bit number j of v_j for all $1 \leq j \leq m$ and return a vector u whose jth bit differs from the jth bit of v_j, for all j. The lower bound is even simpler. If Cantor asks fewer than m queries then there is some vector v_i about which he has no information at the end of the game. In this case he cannot ensure that his vector u will not be equal to this v_i.

Organization. We begin with the case where $m < 2^n$. In the next section (Section 3) we derive nearly tight bounds both in the adaptive and oblivious cases. We do so by

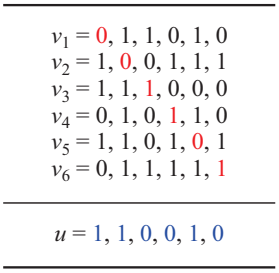

Figure 3. An illustration of Cantor's diagonalization: the vector u at the bottom is not equal to any of the v_i's at the top.

exhibiting and analyzing near optimal strategies for Cantor. Then, in Section 4 we consider the case where $m \geq 2^n$ and derive an optimal bound of $m \cdot n$ in this case (for both the oblivious and the adaptive versions). We do so by exhibiting and analyzing an optimal strategy for Kronecker. Finally, in Section 5 we discuss some algorithmic aspects, and conclude with some suggestions to future research.

3. THE CANTOR-KRONECKER GAME WITH $M < 2^N$.

Adaptive Version

Theorem 1. *Let $g(n, m)$ denote the smallest number of queries that suffices for Cantor when he is allowed to use adaptive strategies. Then,*

$$g(n,m) = \begin{cases} m & m \leq n, \\ 2m - n & n < m < 2^n. \end{cases}$$

Theorem 18 and Remark 19 by Vyas and Williams [**8**] present upper and lower bounds for the adaptive case which are accurate within a multiplicative factor of 2. Theorem 1 closes this gap.

The case $1 \leq m \leq n$ is proved in the previous section so we assume $n \leq m < 2^n$.

Upper Bound. We present a strategy for Cantor which combines diagonalization with another simple idea. To illustrate this idea let us first consider the case $m = n + 1$ for $n \geq 2$. This special case appeared as a question in the *2022 Grossman Math Olympiad* for high-school students, and so perhaps the reader might enjoy trying to solve it before continuing reading.

Let $v_1, \ldots, v_{n+1}$ be the input vectors. Cantor begins with querying the first bit of v_1, v_2, and of v_3. Notice that there must be a bit ε such that at least two vectors among v_1, v_2, v_3 have their first bit equal to ε. Cantor now defines the first bit of u to be $u(1) = 1 - \varepsilon$ and can remove the two vectors among v_1, v_2, v_3 whose first bit equals ε. Now Cantor is left with at most $n - 1$ vectors and can therefore set the last $n - 1$ coordinates of u according to the diagonalization construction.

The general case is handled similarly by induction on n: for $n = 1$ since $n \leq m < 2^n$, m must also be 1 and the result is trivial.

Assuming the result for $n - 1$, let $v_1, \ldots, v_m$ be the m vectors of Kronecker. First, note that there is an integer x satisfying $1 \leq x \leq \lceil m/2 \rceil$ so that $n - 1 \leq m - x < 2^{n-1}$: e.g., when $\lfloor m/2 \rfloor > n - 1$, let $x = \lceil m/2 \rceil$ (thus $m - x = \lfloor m/2 \rfloor$), and when $\lfloor m/2 \rfloor \leq n - 1$, let $x = m - n + 1$ (thus $m - x = n - 1$).

Having x as above, Cantor first queries the first bit of each of the vectors $v_1, v_2, \ldots, v_{2x-1}$. (Note that $2x-1 \leq m$ hence this is possible). There must be a bit $\varepsilon \in \{0,1\}$ so that at least x of the vectors have their first bit equal to ε. Cantor now defines the first bit of his vector u to be $1-\varepsilon$, removes from the set V exactly x of the vectors whose first bit is ε, and defines as V' the set of all restrictions of the remaining $m-x$ vectors to their last $n-1$ coordinates. Note that $n-1 \leq m-x < 2^{n-1}$.

By the induction hypothesis, Cantor can now play the game for the set V' producing an appropriate vector u' by asking at most $2(m-x)-(n-1)$ additional queries. The total number of queries is thus $(2x-1)+2(m-x)-(n-1) = 2m-n$, as needed. The vector u obtained by concatenating the 1-bit vector $1-\varepsilon$ and the vector u' is clearly different from each member of V. This completes the induction step argument and finishes the proof of the upper bound.

Lower Bound. For the lower bound, we present a strategy for Kronecker which essentially mirrors Cantor's strategy from the upper bound. Suppose Cantor manages to produce the required vector u after making exactly b_j queries in coordinate number j of some of the vectors v_i. Kronecker chooses his answers ensuring that for each such j, the answers for bits in the jth location are balanced, that is, at most $\lceil b_j/2 \rceil$ of the answers are 0 and at most $\lceil b_j/2 \rceil$ of the answers are 1.

Consider the vector u produced by Cantor. For every $1 \leq j \leq n$, there are at most $\lceil b_j/2 \rceil$ vectors v_i known to be different than u in coordinate number j. Thus altogether there are at most

$$\sum_{j=1}^{n} \left\lceil \frac{b_j}{2} \right\rceil \leq \sum_{j=1}^{n} \frac{b_j+1}{2}$$

vectors v_i that are known to Cantor to be different than u. In order to ensure u is indeed different from each v_i, this number has to be at least m and hence

$$m \leq \sum_{j=1}^{n} \frac{b_j+1}{2}.$$

By rearranging, this implies that the total number of queries $\sum_{j=1}^{n} b_j$ must be at least $2m-n$. The proof of Theorem 1 is now complete.

Oblivious Version

Theorem 2. *Let $f(n,m)$ denote the smallest number of queries that suffices for Cantor when he is restricted to using oblivious strategies. Then,*

$$f(n,m) = \begin{cases} m & m \leq n \\ m\Big(\log\lceil \frac{m}{n} \rceil \pm O\big(\log\log\lceil \frac{m}{n} \rceil\big)\Big) & n < m < 2^n. \end{cases}$$

Quantitatively, for all $n < m < 2^n$

$$m \cdot \Big(\log\Big(\frac{m}{\ln(2)(n-\log m+1)}\Big) - 1\Big) \leq f(n,m) \leq m\Big\lceil \log \tfrac{2m}{n} + 2\log\big(\log \tfrac{2m}{n}\big) + 1 \Big\rceil.$$

The case $1 \leq m \leq n$ is proved above so we assume $n < m < 2^n$.

Upper Bound. Like in the adaptive case, we present a strategy for Cantor which combines diagonalization with another simple idea. We first illustrate this idea by considering the case $m = n + 1$, and again, we encourage the reader to try and handle this case before continuing reading.

Let $v_1, \ldots, v_{n+1}$ be the input vectors. Cantor begins with querying the first two bits of each of v_1, v_2, and v_3 (for a total of 6 queries). Notice that there are $2^2 = 4$ possible combinations of $0/1$ patterns on the first two bits, but at most three of them are realized by v_1, v_2, v_3. Hence, there must be a pair of bits $\varepsilon_1, \varepsilon_2$ which is not realized by v_1, v_2, nor v_3:

$$(\varepsilon_1, \varepsilon_2) \notin \Big\{ \big(v_1(1), v_1(2)\big), \big(v_2(1), v_2(2)\big), \big(v_3(1), v_3(2)\big) \Big\}.$$

Thus, by setting $u(1) = \varepsilon_1$ and $u(2) = \varepsilon_2$, Cantor rules out v_1, v_2, v_3 and is left with $n - 2$ vectors $v_3, \ldots, v_{n+1}$ which can be obliviously ruled out with $n - 2$ queries using diagonalization.

For the general case, let $1 \leq d \leq n$ be an integer (to be determined later). Pick mutually disjoint subsets of coordinates $J_1, \ldots, J_{\lfloor n/d \rfloor} \subseteq [n]$, each of size d, and pick a partition of the m vectors to $\lfloor n/d \rfloor$ subsets $V_1, \ldots, V_{\lfloor n/d \rfloor}$ such that the partition is as balanced as possible (i.e., the difference between each pair of sizes is ≤ 1). Thus, each set has size

$$|V_i| \leq \left\lceil \frac{m}{\lfloor n/d \rfloor} \right\rceil \leq \frac{2md}{n}.$$

To see why the second inequality holds,
let $x = \frac{n}{d}$ and let $k = \lfloor x \rfloor \geq 1$; thus,

$$\begin{aligned} \left\lceil \frac{m}{\lfloor x \rfloor} \right\rceil = \left\lceil \frac{m}{k} \right\rceil &\leq \frac{m}{k} + \frac{k-1}{k} \\ &\leq \frac{2m}{k+1} \leq \frac{2m}{x}. \end{aligned}$$

The inequality $\frac{m}{k} + \frac{k-1}{k} \leq \frac{2m}{k+1}$ holds since $m > k \geq 1$. Indeed for $m = k + 1$ it holds with equality, and when m is incremented by one and k remains fixed, the left side of the inequality increases by $\frac{1}{k}$, and the right side increases by $\frac{2}{k+1}$, and for $k \geq 1$ we have $\frac{1}{k} \leq \frac{2}{k+1}$. Hence the inequality is valid for $m > k \geq 1$.

Cantor queries (obliviously) as follows.

For each i and each vector in V_i query all the coordinates in J_i.

Thus, the total number of queries is exactly $m \cdot d$. Now, notice that if d satisfies

$$2^d > \frac{2md}{n}, \tag{1}$$

then there must exist an assignment $f_i : J_i \to \{0, 1\}$ such that f_i disagrees with each of the vectors in V_i on at least one coordinate in J_i. Hence Cantor can output the vector u, which agrees with each of the f_i on J_i.

Note that Equation (1) is satisfied exactly when $\frac{2^d}{d} > \frac{2m}{n}$; this inequality holds since $\frac{2^d}{d} \geq y$ when $d \geq \log(y) + 2\log(\log(y)) + 1$ and $y \geq 2$.

Thus for $d = \left\lceil \log\left(\frac{2m}{n}\right) + 2\log\big(\log\big(\frac{2m}{n}\big)\big) + 1 \right\rceil$, the total number of queries is at most

$$m \cdot d = m\left\lceil \log\left(\frac{2m}{n}\right) + 2\log\big(\log\big(\frac{2m}{n}\big)\big) + 1 \right\rceil.$$

Lower Bound. The lower bound proof is based on the following simple idea. Let J_i denote the set of coordinates of v_i which Cantor queries. Thus, the total number of queries Cantor uses is $|J_1| + \ldots + |J_m|$. Now, let $f_i : J_i \to \{0,1\}$ denote Kronecker's answers for the queries on v_i. The crucial observation is that the vector u that Cantor outputs must satisfy

$$(\forall i) : u|_{J_i} \neq f_i.$$

Indeed, if $u|_{J_i} = f_i$ for some i then Kronecker can fail Cantor by picking his ith vector v_i to equal Cantor's output u (which would be consistent with Kronecker's answers).

We summarize the above consideration with a definition that characterizes the winning (or losing) strategies of Cantor in the oblivious case.

Definition 3 (Covering Assignments). We say that a sequence of sets $J_1, \ldots, J_m \subseteq [n]$ *has a covering assignment* if there are m functions $f_i : J_i \to \{0,1\}$ such that every binary vector $v \in \{0,1\}^n$ agrees with f_i on J_i for some $1 \leq i \leq m$ (i.e., $v|_{J_i} = f_i$).

Thus, Kronecker has a winning strategy if and only if the sequence of sets $J_1, \ldots, J_m$ that Cantor queries has a covering assignment. The following lemma establishes the lower bound.

Lemma 4. *Let $J_1, \ldots, J_m \subseteq [n]$ such that*

$$|J_1| + \ldots + |J_m| < m \cdot \Big(\log\big(\frac{m}{\ln(2)(n - \log m + 1)}\big) - 1\Big). \tag{2}$$

Then, $J_1, \ldots, J_m$ has a covering assignment.

Equivalently, if for each vector v_i Cantor queries its entries in J_i and Equation 2 holds, then Kronecker has a winning strategy.

Proof. Let $t_i = |J_i|$ and let $t = \sum_i t_i < m \cdot \Big(\log\big(\frac{m}{n - \log m + 1}\big) - 1\Big)$. Assume, without loss of generality, that $t_1 \leq t_2 \leq \ldots \leq t_m$. We show that there are m functions $f_i : J_i \to \{0,1\}$ so that for every possible vector $v \in \{0,1\}^n$ there is $i \leq m$ so that $v|_{J_i} = f_i$.

We do so by explicitly constructing the f_i's (which corresponds to describing a winning strategy for Kronecker). Starting with the set $V = \{0,1\}^n$ of all possible potential vectors, go over the vectors v_i in order. In step i we choose the function $f_i : J_i \to \{0,1\}$ such that $|\{v \in V : v|_{J_i} = f_i\}|$ is maximized. Since there are 2^{t_i} possible choices for f_i, the maximizing choice satisfies

$$\Big|\{v \in V : v|_{J_i} = f_i\}\Big| \geq \frac{|V|}{2^{t_i}}.$$

After picking f_i, we remove all the vectors of V that agree with f_i and proceed to the next step. Therefore, after the first i steps, the size of the set V of the remaining vectors is at most

$$2^n \prod_{j=1}^{i} (1 - 1/2^{t_j}).$$

These steps are repeated until the size of V shrinks to at most $m/2$, which as we show below happens during the first $\lceil m/2 \rceil$ steps. In each of the remaining steps we simply select f_i which removes at least one vector from V, until we eliminate all of the vectors. This means that if

$$2^n \prod_{j=1}^{\lceil m/2 \rceil} \left(1 - 1/2^{t_j}\right) \leq \frac{m}{2}\,, \tag{3}$$

then the sequence $J_1 \ldots, J_m$ has a covering assignment. So it remains to prove (3).

$$\begin{aligned} 2^n \prod_{j=1}^{\lceil \frac{m}{2} \rceil} \left(1 - \frac{1}{2^{t_j}}\right) &\leq 2^n \prod_{j=1}^{\lceil \frac{m}{2} \rceil} \exp\Big(-\frac{1}{2^{t_j}}\Big) \qquad (1 + x \leq \exp(x) \text{ for all } x \in \mathbb{R}) \\ &= 2^n \exp\Big(-\sum_{j=1}^{\lceil \frac{m}{2} \rceil} \frac{1}{2^{t_j}}\Big) \\ &\leq 2^n \exp\Big(-\frac{m}{2^{\frac{t}{m}+1}}\Big), \end{aligned}$$

where the last inequality follows because $\exp(-x)$ is decreasing and because

$$\sum_{j=1}^{\lceil \frac{m}{2} \rceil} \frac{1}{2^{t_j}} \geq \frac{m}{2} \cdot \frac{1}{2^{\frac{1}{\lceil m/2 \rceil} \sum_{j=1}^{\lceil m/2 \rceil} t_j}} \geq \frac{m}{2} \cdot \frac{1}{2^{\frac{t}{m}}},$$

which follows by convexity of the function $f(x) = 2^x$ and because $t_1 \leq t_2 \leq \ldots \leq t_m$.

We have thus shown that if $|J_1| + \ldots + |J_m| = t$, where t is such that

$$2^n \exp\Big(-\frac{m}{2^{\frac{t}{m}+1}}\Big) \leq \frac{m}{2}\,,$$

then the sequence $J_1, \ldots, J_m$ has a covering assignment. The last inequality surely holds provided

$$\frac{m}{2^{\frac{t}{m}+1}} \geq \ln(2)(n + 1 - \log m).$$

That is, the inequality holds provided

$$2^{\frac{t}{m}+1} \leq \frac{m}{\ln(2)(n + 1 - \log m)},$$

or

$$t \leq m \cdot \Big(\log\big(\frac{m}{\ln(2)(n + 1 - \log m)}\big) - 1\Big)$$

which is guaranteed by the premises of the lemma, thus completing the proof. ∎

4. THE CANTOR-KRONECKER GAME WITH $M \geq 2^N$. Assume now that Kronecker's list V consists of $m \geq 2^n$ binary vectors of length n. In this case V may contain all the binary vectors of length n and there may be no vector Cantor can output that is different from each vector on Kronecker's list. In this regime it is more natural to first focus on the decision problem in which Cantor's goal is to decide whether V contains $\{0,1\}^n$, and if this is not the case, to provide a vector which is not in V.[2] Clearly Cantor can achieve this if he queries all mn entries. Can he do better?

We first observe that mn queries are in fact needed in the oblivious case: assume that Cantor submits only $mn-1$ queries, and leaves the jth bit of v_i unqueried. Then Kronecker may set v_i to be the unique occurrence of the all ones vector 1^n, and set the remaining $m-1$ vectors in V to include all 2^n-1 vectors that are different from the all ones vector. Clearly, it is necessary for Cantor to query also the last bit of v_i in order to see whether v_i is the all ones vector or not. Consequently, Cantor must query all mn queries in the oblivious case.

How about the adaptive case? The following (similar to the above) argument shows that for $m = 2^n$, Kronecker can force $mn = 2^n n$ queries in the adaptive case. Use any list which contains each binary vector of length n exactly once. After $mn-1$ bits are queried, the last unqueried bit belongs to a vector which occurs only once in V. Assume without loss of generality that this bit is 0. If this bit is set to 1, then V does not contain all 2^n binary vectors of length n. Hence it is necessary to get the value of this last bit.

The case when $m > 2^n$ turns out to be more subtle. Nevertheless, we prove that mn queries are necessary even in this case. We start with introducing some notation.

Notation. Each step of the game consists of a query by Cantor followed by a response by Kronecker. The status of the game after each such step is given by an $m \times n$ matrix L, where $L(i,j)$ denotes the status of the jth bit of v_i, that is: $L(i,j) \in \{0,1,\star\}$, where $L(i,j) = \star$ means that the jth bit of v_i has not been queried yet, and otherwise $L(i,j)$ equals the value of this bit as answered by Kronecker.

Definition 5. $\texttt{FIXED}(L) = \left\{v \in L : v \in \{0,1\}^n\right\}$. That is, $\texttt{FIXED}(L)$ is the set of all vectors in L that were fully queried by Cantor.

Definition 6. L is *complete* if $\texttt{FIXED}(L) = \{0,1\}^n$.

Definition 7. A subset S of 2^n rows of L is *useful* if it either contains all the 2^n binary vectors of length n, or it can be *converted* to this set by replacing each $\star$-entry in S by 0 or 1.

Definition 8. A matrix L is *unblocked* if it can be completed; that is, if L has a useful subset. Otherwise L is called *blocked*.

Notice that for $m \geq 2^n$, the m by n matrix all whose entries are $\star$ is unblocked.

As a warmup, and to get used to the definitions, let us assume first that Cantor queries the vectors one by one according to their order. That is, he first queries all the bits of v_1 from left to right, then all the bits of v_2 from left to right and so on. We use the following strategy for Kronecker: when Cantor queries the jth bit of v_i (i.e., the value of $L(i,j)$), Kronecker replies according to the following "0 first" strategy.

$$\text{Modified value of } L(i,j) = \begin{cases} 1 & \text{If setting } L(i,j) \text{ to 0 blocks L.} \\ 0 & \text{Otherwise.} \end{cases} \tag{4}$$

[2] Our results below imply that the decision and search variants are equivalent for $m \geq 2^n$ (for $m < 2^n$ the decision variant is trivial), in the sense that in both cases it is necessary and sufficient to query all $m \cdot n$ entries.

It is not hard to verify that since Cantor queries the vectors one by one and from left (most significant bit) to right, the following matrix is produced. Each of the first $m - 2^n + 1$ rows will be set to the all-zeros vector, and the last $2^n - 1$ rows will be set to the $2^n - 1$ nonzero vectors in increasing lexicographical order: starting with $0^{n-1}1$ and ending with 1^n. Hence Cantor is forced to query all mn entries as in the oblivious case.

It turns out that, for *any* strategy of Cantor, the above "0 first" strategy of Kronecker forces Cantor to make mn queries.

Theorem 9. *Let $m > 2^n$. Then for any strategy of Cantor, the "0 first" strategy of Kronecker forces Cantor to make mn queries in order to determine if L contains $\{0,1\}^n$.*

In the following we consider an arbitrary execution of the game, where Kronecker follows the "0 first" strategy (and Cantor's strategy is arbitrary). We denote by L_t the $m \times n$ matrix L after t steps of the game. Thus L_0 is the initial matrix which is filled only with $\star$'s. Notice that if L is unblocked and $L(i,j) = \star$, then either setting $L(i,j)$ to 0 or setting $L(i,j)$ to 1 does not block L. Therefore

Observation 10. *If L_t is unblocked, so is L_{t+1}. Hence L_{mn} is complete; i.e., it contains $\{0,1\}^n$.*

Definition 11. We say that a row $L(i)$ is *essential* for an unblocked matrix L if every useful subset of L's rows contains $L(i)$.

Note that if $L_t(i)$ is essential for L_t, then $L_s(i)$ is essential for L_s for all $s \geq t$. Also, if $L_{mn}(i)$ is essential for L_{mn}, then $L_{mn}(i)$ is equal to a unique vector in $\{0,1\}^n$ which is different from all other rows of L_{mn}.

Lemma 12. *Assume that $L_t(i)$ is not essential for L_t and $L_t(i,j) = \star$. If $L_t(i,j)$ is queried at time $t+1$, then it is set to 0, i.e., $L_{t+1}(i,j) = 0$.*

Proof. If $L(i)$ is not essential for an unblocked matrix L, then $L(i,j)$ can be set to any bit without blocking $L(i)$. Hence, by the '0 first" strategy, it is set to 0. ∎

By a straightforward induction Lemma 12 implies

Corollary 13. *If $L_t(i)$ is not essential for L_t, then $L_t(i)$ contains no 1's (only 0's or $\star$'s). In particular, if $L_{mn}(i)$ is not essential for L_{mn}, then $L_{mn}(i)$ is the zero vector 0^n. Hence, every row of L_{mn} which is not the zero vector is essential, and thus it is different from all other rows of L_{mn}.*

Lemma 14. *Let $L_{mn-1}(i,j)$ be the last bit queried in the game. Then $L_{mn-1}(i)$ is an essential row of L_{mn-1}.*

Proof. To simplify notation, we assume without loss of generality that $j = 1$. Assume towards contradiction that $L_{mn-1}(i)$ is not essential for L_{mn-1}. By Corollary 13, this implies that $L_{mn-1}(i) = \star 0^{n-1}$ and $L_{mn}(i) = 0^n$. (i.e., Kronecker sets $L_{mn-1}(i,1)$ to 0 at Cantor's mn'th query). Since L_{mn} is complete (Observation 10), this implies that L_{mn-1} contains a distinct occurrence of each of the $2^n - 1$ nonzero vectors of $\{0,1\}^n$, and in particular for some $k \neq i$, $L_{mn-1}(k)$ is the unique row of L_{mn-1} which equals 10^{n-1}. Then, any subset S of L_{mn-1} which contains

- the row $L_{mn-1}(i)$,

- the $2^n - 2$ nonzero rows of L_{mn-1} excluding $L_{mn-1}(k)$, and
- some zero row of L_{mn-1} (by Corollary 13 there are $m - 2^n > 0$ such rows in L_{mn-1}),

is a useful subset of L_{mn-1} which does not contain $L_{mn-1}(k)$. Hence $L_{mn-1}(k)$ is not essential for L_{mn-1}, and by Lemma 12 $L_{mn-1}(k, 1) = 0 \neq 1$, which stands in contradiction with $L_{mn-1}(k) = 10^{n-1}$. ■

Proof of Theorem 9. Let $L_{mn-1}(i, j)$ be the last query in the game. By Lemma 14, $L_{mn-1}(i)$, and hence also $L_{mn}(i)$, is essential, meaning that $L_{mn}(i)$ is different from all other rows of L_{mn}. Thus Cantor must get the value of $L_{mn-1}(i, j)$ in order to reach a decision. ■

A remark on computational complexity. A naive implementation of the "0 first" strategy might take exponential time. Indeed, the naive implementation involves checking whether setting the queried bit to 0 blocks the current matrix, which involves checking a potentially exponential list of constraints. Nevertheless, we next show that this strategy in fact admits a polynomial-time implementation. First, notice that the initial $m - 2^n$ steps are trivially efficient, because setting $L(i, j)$ to any value cannot block L (since at least 2^n rows of L are not queried yet).

Thus it suffices to show that each later step can be performed in time which is polynomial in mn, the size of L. In other words, deciding whether setting $L(i, j)$ to 0 blocks the matrix, can be performed in polynomial time. Let L_t be the matrix L after t steps of the game, $t > m - 2^n$. Consider the bipartite graph $G_t = (A_t, B, E_t)$, where $A_t = \{L_t(i) : 1 \leq i \leq m\}$ is the set of rows of L_t, $B = \{0, 1\}^n$, and $(L_t(i), u) \in E_t$ if and only if $L_t(i)$ can be converted to the binary vector u by replacing the $\star$'s in $L_t(i)$ (if any) by binary digits. Then, a subset S of L_t is useful for L_t if and only if G_t contains a perfect matching between the vertices in A_t which correspond to S and B.

Assume now that we are given the graph G_t, and some perfect matching M_t for G_t as above, and let $L_t(i, j)$ be the entry queried by Cantor at step $t + 1$. To check if setting $L_t(i, j)$ to 0 blocks L_t, we remove from G_t all the edges $(L_t(i), u)$ in which $u(j) = 1$, and check if the resulted graph contains a perfect matching. Recall that we are given a perfect matching M_t for G_t, and removing these edges eliminates at most one edge from M_t. If no edge of M_t was removed then we are done. Otherwise, this checking can be done by searching an augmenting path in the resulted graph. This can be accomplished in $O(|E_t|) = O(m2^n) = O(m^2)$ time by executing one phase in some classical algorithm for bipartite matching (see, e.g., [**4**]).

5. CONCLUDING REMARKS AND FUTURE RESEARCH. We studied the Cantor-Kronecker game for different values of m and n. When $m \leq n$ the trivial lower bound of m is tight (a lower bound of m follows because Cantor must query at least one bit in each vector); when $m \geq 2^n$, the trivial upper bound of mn is tight (an upper bound of mn follows because querying all the bits is clearly sufficient); when $n < m < 2^n$ the landscape is more interesting, and in particular the bounds depend on whether Cantor is adaptive or oblivious.

Further Research. We conclude with suggestions for possible future research:

1. Study the Cantor-Kronecker game when there are r rounds of adaptivity. There are r rounds in which Cantor can submit queries, and in each round the submitted queries may depend on Kronecker's answers to queries from previous rounds. How does the

query complexity change as a function of r? Note that $r = 1$ is the oblivious case and $r = \infty$ is the adaptive case.[3]

2. It is also natural to consider randomized variants of the game. Can Cantor use fewer queries if he is allowed to use randomness? In this context it is natural to assume that Kronecker picks the vectors v_i's before the game begins, and Cantor's goal is to minimize the expected number of queries until finding a missing vector u. Alternatively, one can allow Cantor a small error. That is, with probability at most ε Cantor can output a vector u which is equal to one of the v_i's. Note that ε should be smaller than $\frac{m}{2^n}$ or else the problem becomes trivial since then Cantor can output a vector u which is sampled uniformly and be successful with probability at least $1 - \varepsilon$ without submitting even a single query.
3. Consider the following generalization of the game. Let $k \leq m, \ell \leq n$ be positive integers. Kronecker maintains an $m \times n$ binary matrix, and Cantor queries the entries of Kronecker's matrix. Cantor's goal is to find a $k \times \ell$ matrix which does not appear as a submatrix of Kronecker's $m \times n$ matrix, or to decide that one does not exist. So, the original game is when $k = 1, \ell = n$. What is the query complexity as a function of k, ℓ, m, n in the adaptive/oblivious case? For which values does Cantor have a strategy that uses strictly less than $m \cdot n$ queries?
4. Find tighter bounds for the oblivious case. Specifically, notice that Cantor's original diagonalization provides tight bounds on the number of queries needed for the oblivious case when $m \leq n$. It will be interesting to derive tight bounds and optimal strategies in the remaining cases. As we show below, this question has connections with natural combinatorial problems.

   Consider the case when m is at the other end of the scale, namely $2^{n-1} \leq m < 2^n$. Then, Cantor can win the game by querying $mn - d$ bits, where $d = 2^n - m - 1$. In fact, it suffices that Cantor chooses his queries such that each of the d unqueried entries belongs to a different vector. In this case any assignments of values to the unqueried entries covers (in the sense of Definition 3) the $m - d$ fully queried vectors, and at most two additional vectors per each of the remaining d vectors (each of which contains one unqueried entry); altogether at most $(m - d) + 2d = m + d$ vectors. Hence, Cantor is guaranteed to win the game provided that $m + d < 2^n$ (equivalently $d \leq 2^n - m - 1$).

   Is the above strategy optimal? That is, can Kronecker win the game when Cantor queries only $mn - (2^n - m)$ bits? Informally, Kronecker has a winning strategy if, for any distribution of the $2^n - m$ unqueried entries, there is an assignment which covers *sufficiently many* vectors, where "sufficiently many" is detailed below.

Definition 15 (`cube`$(v), J$**-cube).** Let v be a vector with possibly some unqueried entries. `cube`(v) is the set of binary vectors which can be obtained by replacing the unqueried entries in v by zeros or ones. In particular, `cube`$(v) = \{v\}$ if v is fully queried. The cube `cube`(v) is called a J-cube if $J = \{j :$ the jth bit of v is not queried$\}$. For $j \in [n]$, a $\{j\}$-cube is denoted by j-edge.

Assume that Cantor distributes the $(2^n - m)$ unqueried entries among vectors $v_1, \ldots, v_q$. Then Kronecker's answers to the queried entries define a cube $C(v_i)$ for each vector v_i. Kronecker wins if and only if those cubes cover $\{0, 1\}^n$. Hence Kronecker has a winning strategy when Cantor uses $mn - (2^n - m)$ queries ($2^{n-1} + 1 \leq m < 2^n$) if and only if the following holds.

[3] In fact $r = n$ is already equivalent to $r = \infty$ because the optimal strategy presented in the proof of Theorem 1 uses n rounds of adaptivity.

Conjecture 16. *Let $d = 2^n - m < 2^{n-1}$. For any collection $J_1, J_2, \ldots, J_q$ of nonempty subsets of $[n]$ satisfying $\sum_{i=1}^{q} |J_i| = d$, there are cubes $C_1, \ldots, C_q$ such that C_i is a J_i-cube, and $|\bigcup_{i=1}^{q} C_i| \geq d + q$.*

The following result of [**5**] proves Conjecture 16 for the case that each J_i-cube is a singleton $\{j_i\}$.

Theorem 17 ([5]). *Let $d < 2^{n-1}$. For any multiset $D = \{j_1, j_2, \ldots, j_d\}$ of elements of $[n]$, $\{0,1\}^n$ contains a matching $\{e_1, \ldots, e_d\}$ such that for $i = 1, \ldots, d$, e_i is a j_i-edge.*

It is also shown in [**5**] that the conclusion of Theorem 17 (and hence also the conclusion of Conjecture 16) does not hold when $d = 2^{n-1}$. In this case a corresponding matching exists if and only if each element in $[n]$ occurs an even number of times in D.

This implies that when $m = 2^{n-1}$ Cantor has a winning strategy with only $mn - (2^n - m) = mn - 2^{n-1}$ queries. He may query $n - 1$ entries for each vector, so that at least one dimension is left unqueried in an odd number of vectors.

5. Finally, Vyas and Williams [**8**] investigate a local variant of the game in which Cantor's goal is not to output a full vector u, but only to output a pre-specified bit, u_j, of it. That is, Cantor is given as an input an entry j and should make as few queries as possible and output the bit u_j. Notice that when the number of vectors m satisfies $m \leq n$ then the basic diagonalization strategy queries only a single entry, namely $v_j(j)$. Vyas and Williams show that then at least $\Omega(\frac{m}{n})$ queries are necessary and at most $O(\frac{m}{n} \log \frac{m}{n})$ queries are sufficient. However, the precise relationship between m and n that minimizes the number of queries is left as an interesting open question.

Acknowledgements. We would like to thank Nikhil Vyas and Ryan Williams for bringing references [**1, 7, 8**] to our attention, and Ron Holzman for informing us about the result in [**5**]. We also thank Ariel Gabizon and Yuval Wigderson for providing insightful comments on a previous version of this manuscript. Finally we would like to thank the anonymous referees for their very helpful comments.

The research of Noga Alon was supported by NSF grant DMS-2154082. The research of Kasper Green Larsen was supported by a DFF Sapere Aude Research Leader Grant No. 9064-00068B. The research of Shay Moran is supported by a Robert J. Shillman Fellowship, by ISF grant 1225/20, by BSF grant 2018385, by an Azrieli Faculty Fellowship, by Israel PBC-VATAT, by the Technion Center for Machine Learning and Intelligent Systems (MLIS), and by the the European Union (ERC, GENERALIZATION, 101039692). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

REFERENCES

1. Barzdin, J. M., Freivald, V. (1972). On the Prediction of General Recursive Functions (in Russian). *Soviet Math. Doklady*. 13: 1224–1228.
2. Cantor, G. (1874). Über eine Eigenschaft des Inbegriffs aller reellen algebraischen Zahlen. *J. Reine Angew. Math.* 1(77): 258–262.
3. Dauben, J. W. (1982). *Georg Cantor: His Mathematics and Philosophy of the Infinite*. Princeton, NJ: Princeton University Press.
4. Even, S. (2011). *Graph Algorithms*, 2nd edition. New York, NY: Cambridge University Press.
5. Felzenbaum, A., Holzman, R., Kleitman, D. J. (1993). Packing Lines in a Hypercube. *Disc. Math.*. 117(1): 107–112.
6. Gamow, G. (1988). *One, Two, Three–Infinity: Facts and Speculations of Science*. New York: Dover Books on Mathematics Series.

7. Kannan, R. (1982). Circuit-Size Lower Bounds and Non-Reducibility to Sparse Sets. *Inf. Control*. 55(1): 40–56.
8. Vyas, N., Williams, R. (2023). On Oracles and Algorithmic Methods for Proving Lower Bounds. *14th ITCS*.
9. Wigderson, A. (2019). *Mathematics and Computation: A Theory Revolutionizing Technology and Science*. Princeton, NJ: Princeton University Press.

NOGA ALON is a Professor of Mathematics at Princeton University and a Professor Emeritus of Mathematics and Computer Science at Tel Aviv University, Israel. He works in Discrete Mathematics and its applications in Theoretical Computer Science, Information Theory, Combinatorial Geometry, and Combinatorial Number Theory. He is a member of the Israel Academy of Sciences and Humanities and of the Academia Europaea, and an honorary member of the Hungarian Academy of Sciences. He received several awards, two recent ones are the 2022 Shaw Prize in Mathematical Sciences and the 2022 Knuth Prize for outstanding contributions to the foundations of computer science.
Department of Mathematics, Princeton University and Departments of Mathematics and Computer Science, Tel Aviv University.
nalon@math.princeton.edu

OLIVIER BOUSQUET received his PhD in Applied Mathematics from Ecole Polytechnique, France in 2002. He was then a researcher at the Max Planck Institute in Tübingen, working on Machine Learning and in particular Statistical Learning Theory and Kernel Methods. In 2004 he joined a startup company where he led a research team and developed ML software for predicting manufacturing quality. Olivier joined Google in 2007 and contributed to many aspects of the search engine, in particular leading an engineering team working on Language Understanding and the Knowledge Graph. In 2016, he joined the Research team and has been working on Deep Learning and Foundation Models. His research interests include theoretical aspects of Machine Learning with a focus on understanding the phenomenon of generalization from a statistical and combinatorial perspective.
Google DeepMind Zürich
obousquet@google.com

KASPER GREEN LARSEN is a Professor of Computer Science at Aarhus University. His research interests include theoretical aspects of machine learning, algorithms, data structures, complexity theory, and cryptography. He has received best paper and best student paper awards at the premier theoretical computer science conferences, STOC and FOCS, as well as the top cryptography conference, CRYPTO, and the top machine learning theory conference, COLT. He is also the recipient of the Presburger Award (2019).
Department of Computer Science, Aarhus University.
larsen@cs.au.dk

SHAY MORAN is an Associate Professor at the Technion and a researcher at Google Tel-Aviv. His research focuses on combinatorial and geometric problems that are inspired by computer science with a focus on machine learning theory. Shay's work has been recognized with best paper awards in premier theoretical computer science conferences such as FOCS and COLT and he received several prestigious prizes and grants, including the Azrieli Faculty Fellowship (2020-2024), Cooper Award (2022), Krill Prize (2023), and an ERC starting grant (2022-2027).
Departments of Mathematics, Computer Science, and Data and Decision Sciences, Technion, Israel
smoran@technion.ac.il

SHLOMO MORAN is a Professor Emeritus at the Department of Computer Science, Technion, Israel. His research involves various theoretical aspects of computer science, including approximation algorithms for NP hard problems, complexity theory, distributed algorithms, fault tolerance of asynchronous systems, self stabilizing algorithms, search engine algorithms, reconstructing of phylogenetic (evolutionary) trees, dynamic networks, and more.
Department of Computer Science, Technion, Israel.
moran@cs.technion.ac.il