

Directed Cut-Query Complexity of Finding a King

Ziad Ismaili Alaoui ^{*}†

Oded Lachish [‡]

Nikhil S. Mande ^{*}

Abstract

A king in a tournament is a vertex from which every other vertex is reachable by a directed path of length at most two. We study the deterministic query complexity of finding a king when a query specifies a set of vertices and returns the number of edges directed from that set to its complement. We prove that this complexity is $\Theta(n)$ for n -vertex tournaments. To show this, we first show that every correct deterministic algorithm in this model must output a maximum-outdegree vertex, even though a king need not have maximum outdegree. We then give an adversary whose answers, after too few queries, remain consistent with a tournament in which the algorithm’s output does not have maximum outdegree.

1 Introduction

A tournament is an orientation of a complete graph. A vertex is a *king* if it can reach every other vertex by a directed path of length at most two. It is now a well-known fact [Lan53] that every tournament has a king: any vertex of maximum outdegree is one. Finding such a vertex is a basic search problem when the directions of the edges are initially unknown.

In the standard edge-query model, a query reveals the direction of one edge, and an algorithm’s goal is to find a king with as few queries as possible in the worst case. Shen, Sheng, and Wu [SSW03] proved an $\Omega(n^{4/3})$ lower bound and an $O(n^{3/2})$ upper bound for finding a king. Later work studied finding variants of kings [BJRS22, LRT22, IM25], randomized and quantum algorithms [MPS23], and the problem of finding multiple kings [AGNS24]. Mande, Paraashar, Sanyal, and Saurabh [MPSS24] studied the communication complexity of finding a king when the edges of the tournament are partitioned between two players.

Cut queries have been extensively studied as a model for solving graph problems without learning the entire graph. In an undirected graph, a cut query specifies a vertex set S and returns the number of edges between S and its complement. The connection to value-oracle submodular function minimization, together with the possibility of solving graph problems without reconstructing the input, has motivated the study of cut-query algorithms for a range of problems [RSW18, MN20, AEG⁺22, CL23, ASW25, KM26].

In the directed cut-query model, a query specifies a vertex set S and returns the number of edges directed from S to its complement. A central question is which problems can be solved using substantially fewer queries than the number of edges. This is particularly relevant for tournaments, which have $\Theta(n^2)$ edges. Ghosh, Kuchlous, Mehra, and Mukhopadhyay [GKMM26] gave a unified framework for solving tournament problems from the vertex degrees, encompassing earlier results on reachability and strong connectivity and yielding new algorithms for minimum cut. Their framework gives $O(n)$ -query algorithms for these problems in the directed cut-query model. We study the directed cut-query complexity of finding a king.

Formally, given a tournament T on the known vertex set $[n]$, a directed cut query specifies a set $S \subseteq [n]$ and returns

$$c_T(S) = |\{(u, v) \in S \times ([n] \setminus S) : u \rightarrow v \text{ in } T\}|.$$

^{*}University of Liverpool, United Kingdom. {ziad.ismaili-alaoui, nikhil.mande}@liverpool.ac.uk

[†]Philipps-Universität Marburg, Germany.

[‡]University of London, Birkbeck, United Kingdom. o.lachish@bbk.ac.uk

Ziad Ismaili Alaoui was funded by EPSRC grant EP/X039447/1.

Each query has unit cost, and queries may be chosen adaptively. We write $D_{\text{dc}}(\text{KING}_n)$ for the minimum worst-case number of queries required by a deterministic algorithm that outputs a king of every n -vertex tournament.

The following is our main result in this paper.

Theorem 1. *For all sufficiently large n ,*

$$\frac{n}{3} < D_{\text{dc}}(\text{KING}_n) \leq n - 1.$$

Our proof in fact gives a lower bound of $(1/2 - \epsilon)n$ for every constant $\epsilon > 0$ and all sufficiently large n .

1.1 Proof idea

For the upper bound, we query the outdegrees of $n - 1$ vertices, using one singleton cut query for each vertex. Since the sum of all outdegrees is $\binom{n}{2}$, we can infer the remaining vertex's outdegree. We then return a maximum-outdegree vertex, which is always a king [Lan53].

For the lower bound, we first observe that the answer to any directed cut query is determined by the outdegree of each vertex. Thus an algorithm cannot distinguish two tournaments in which each vertex has the same outdegree in both tournaments. We show that if a vertex does not have maximum outdegree, then the edges can be reoriented, preserving every vertex's outdegree, so that this vertex is no longer a king. Consequently, every correct deterministic cut-query algorithm must output a maximum-outdegree vertex.

It remains to show that finding a maximum-outdegree vertex requires $\Omega(n)$ queries. We show this by an adversarial argument, answering queries as if every vertex had outdegree $(n - 1)/2$. Towards a contradiction, suppose a cheap algorithm was correct and outputted vertex v against this adversary. A counting argument shows that, after too few queries, we can perturb the degrees so that they are realized by a tournament consistent with all answers given so far in which another vertex has strictly larger outdegree than that of v . The algorithm also outputs v on this tournament, contradicting our earlier conclusion that every correct deterministic cut-query algorithm must output a maximum-outdegree vertex.

Connection with communication complexity. Mande, Paraashar, Sanyal, and Saurabh [MPSS24] proved an $\Omega(n)$ deterministic communication lower bound for finding a king when the edges are partitioned between two players. Since each directed cut query can be simulated using $O(\log n)$ bits of communication, this implies an $\Omega(n/\log n)$ lower bound on the directed cut-query complexity of finding a king. They also proved an $\Omega(n \log n)$ deterministic communication lower bound for finding a maximum-outdegree vertex. Our result that every correct deterministic cut-query algorithm must output a maximum-outdegree vertex therefore gives an alternative proof of the $\Omega(n)$ lower bound. We give the details in Section 4.

2 Equivalence of king-finding and maximum-outdegree search

We first show that directed cut queries reveal no information beyond the outdegree of each vertex. For a tournament T and a vertex v , let $d_T^+(v)$ denote the outdegree of v in T .

Lemma 2. *Let T_1 and T_2 be tournaments on the same labelled vertex set V , with $d_{T_1}^+(v) = d_{T_2}^+(v)$ for every $v \in V$. Then every deterministic directed cut-query algorithm has the same execution and output on T_1 and T_2 .*

Proof. The sum $\sum_{v \in S} d_T^+(v)$ counts every edge whose origin lies in S . These are precisely the edges directed from S to its complement and the edges within S , each counted once. Since T is a tournament, there are exactly $\binom{|S|}{2}$ edges within S . Therefore,

$$c_T(S) = \sum_{v \in S} d_T^+(v) - \binom{|S|}{2}. \quad (1)$$

Since $d_{T_1}^+(v) = d_{T_2}^+(v)$ for every $v \in V$, the right-hand side is the same for T_1 and T_2 for every choice of S . Hence $c_{T_1}(S) = c_{T_2}(S)$ for every $S \subseteq V$. Since the algorithm is deterministic, it asks the same queries and produces the same output on both tournaments. $\square$

For a tournament T on vertex set $[n]$, we write

$$\mathbf{d}(T) = (d_T^+(1), \dots, d_T^+(n)).$$

This is a *labelled* vector: its i th coordinate belongs to vertex i . Equality of vectors always means coordinate-wise equality, not equality up to permutation. A vector is *realizable* if it equals $\mathbf{d}(T)$ for some tournament T on $[n]$. We write $N_T^+(v)$ for the out-neighbours of v , and $N_T^+[v] = N_T^+(v) \cup \{v\}$. Subscripts are omitted when the tournament is clear.

Definition 3. We say that a vertex w covers v if $w \rightarrow v$ and $w \rightarrow x$ for every $x \in N^+(v)$.

We make the simple observation that if a vertex is covered, then it cannot be a king. This is because if w covers v , then there is no path of length two from v to w .

Observation 4. If a vertex w covers another vertex v in a tournament T , then v is not a king in T .

Conversely, if v is not a king, choose a vertex w that it cannot reach in at most two steps. Then w covers v . In particular, if w covers v , then $d^+(w) \geq d^+(v) + 1$. Consequently,

$$v \text{ is not a king} \implies \exists w \text{ such that } d^+(w) \geq d^+(v) + 1. \quad (2)$$

In particular, the contrapositive of the above recovers the fact that every maximum-outdegree vertex is a king [Lan53].

We next show that any vertex whose outdegree is not maximum can be made a non-king by reorienting edges while preserving every vertex's outdegree. We will use the following observation.

Observation 5. For any directed triangle in a tournament, reversing all of its three edges preserves the outdegree of every vertex.

Proof. Each vertex of the triangle has exactly one outgoing edge within the triangle both before and after the reversal. All other edges are unchanged. $\square$

Lemma 6. Let $\mathbf{d} = (d_1, \dots, d_n)$ be a realizable degree vector, and let $v, w \in [n]$ satisfy $d_w > d_v$. There is a tournament T with $\mathbf{d}(T) = \mathbf{d}$ in which w covers v .

Proof. Start with any realization T of $\mathbf{d}$. We first arrange that $w \rightarrow v$. If instead $v \rightarrow w$, there is a vertex y with $w \rightarrow y \rightarrow v$: otherwise $N^+(w) \subseteq N^+(v)$, contradicting $d_w > d_v$. Reverse the directed triangle $v \rightarrow w \rightarrow y \rightarrow v$. By Observation 5, the degrees are preserved and we now have the edge $w \rightarrow v$. With $w \rightarrow v$ fixed, let

$$A = N^+(v) \setminus N^+(w), \quad B = N^+(w) \setminus (N^+(v) \cup \{v\}).$$

Writing $C = N^+(v) \cap N^+(w)$, we have $d_v = |C| + |A|$ and $d_w = |C| + |B| + 1$. Hence

$$|B| - |A| = d_w - d_v - 1 \geq 0. \quad (3)$$

If A is nonempty, then B is also nonempty, so we may choose $x \in A$ and $y \in B$. The definitions give $v \rightarrow x \rightarrow w \rightarrow y \rightarrow v$. There are two cases to consider:

- If $x \rightarrow y$, reverse the triangle $v \rightarrow x \rightarrow y \rightarrow v$. Vertex v loses x and gains y as an out-neighbour, while $N^+(w)$ is unchanged. Since $y \in N^+(w)$, the new set A is exactly $A \setminus \{x\}$.
- If $y \rightarrow x$, reverse the triangle $w \rightarrow y \rightarrow x \rightarrow w$. Vertex w loses y and gains x as an out-neighbour, while $N^+(v)$ is unchanged. Since $y \notin N^+(v)$, again the new set A is exactly $A \setminus \{x\}$.

In both cases every degree is preserved, the edge $w \rightarrow v$ is unchanged, and $|A|$ decreases by one. Redefining A, B, C using the updated tournament, we again have $d_v = |C| + |A|$ and $d_w = |C| + |B| + 1$. Thus $|B| - |A| = d_w - d_v - 1 \geq 0$ by Equation (3), so whenever A is nonempty, B is also nonempty and another step is possible. This process therefore terminates with $A = \emptyset$, giving $N^+(v) \subseteq N^+(w)$. Together with $w \rightarrow v$, this means that w covers v , and the resulting tournament still satisfies $\mathbf{d}(T) = \mathbf{d}$. $\square$

Corollary 7. *For every realizable degree vector $\mathbf{d}$, the vertices that are kings in every realization of $\mathbf{d}$ are exactly the vertices of maximum outdegree in $\mathbf{d}$.*

Proof. A maximum-outdegree vertex is a king by Equation (2). Every other vertex is covered in some realization by Lemma 6, and hence not a king by Observation 4. $\square$

Let MOD_n denote the task of finding a maximum-outdegree vertex.

Corollary 8. *Every correct deterministic directed cut-query algorithm for KING_n outputs a maximum-outdegree vertex on every input. Consequently,*

$$D_{\text{dc}}(\text{KING}_n) = D_{\text{dc}}(\text{MOD}_n).$$

Proof. Let A be a correct deterministic directed cut-query algorithm for KING_n . Suppose that on some input tournament T , it outputs a vertex v that does not have maximum outdegree. By Corollary 7, there exists a tournament T' with $\mathbf{d}(T') = \mathbf{d}(T)$ such that v is not a king in T' . By Lemma 2, the output of A on T' is the same as its output on T , namely v . This contradicts the correctness of A .

Thus every correct deterministic algorithm for KING_n also solves MOD_n , which implies $D_{\text{dc}}(\text{KING}_n) \geq D_{\text{dc}}(\text{MOD}_n)$. The reverse inequality follows because every maximum-outdegree vertex is a king by Equation (2). $\square$

Proposition 9. *For every $n \geq 1$, $D_{\text{dc}}(\text{KING}_n) \leq n - 1$.*

Proof. Query $c_T(\{v\}) = d_T^+(v)$ for $n - 1$ vertices. Infer the remaining degree from $\sum_v d_T^+(v) = \binom{n}{2}$, and output a vertex of maximum outdegree. $\square$

3 A direct lower bound

In this section we first note sufficient conditions for a degree vector to be realizable, and then give the counting argument used to preserve the query answers.

3.1 Realizing bounded degree perturbations

Theorem 10 (Landau [Lan53], [Moo68, Theorem 29]). *Let $d_1 \leq \dots \leq d_n$ be nonnegative integers. The vector $\mathbf{d} = (d_1, \dots, d_n)$ is realizable if and only if*

$$\sum_{i=1}^k d_i \geq \binom{k}{2} \quad (1 \leq k \leq n), \quad \sum_{i=1}^n d_i = \binom{n}{2}.$$

Lemma 11. *Let $n \geq 4$ and $\mathbf{d} \in \mathbb{Z}^n$ satisfy*

$$\frac{n}{4} \leq d_i \leq \frac{3n}{4} \quad (i \in [n]), \quad \sum_{i=1}^n d_i = \binom{n}{2}.$$

Then $\mathbf{d}$ is realizable.

Proof. Sort the coordinates into nondecreasing order. For $k \leq n/2 + 1$,

$$\sum_{i=1}^k d_i \geq \frac{nk}{4} \geq \frac{k(k-1)}{2} = \binom{k}{2}.$$

For $k \geq n/2 + 1$, the bound on the remaining coordinates gives

$$\begin{aligned} \sum_{i=1}^k d_i - \binom{k}{2} &= \sum_{i=1}^n d_i - \sum_{i=k+1}^n d_i - \binom{k}{2} \geq \binom{n}{2} - \frac{3n}{4}(n-k) - \binom{k}{2} \\ &= \frac{n(n-1) - k(k-1)}{2} - \frac{3n}{4}(n-k) \\ &= \frac{(n-k)(n+k-1)}{2} - \frac{3n}{4}(n-k) = \frac{(n-k)(2k-n-2)}{4} \geq 0, \end{aligned}$$

where the last inequality holds since we are in the case where $k \geq n/2 + 1$. Thus the conditions in Theorem 10 are satisfied, implying a realization of the sorted vector, and hence the original vector by relabeling vertices. $\square$

3.2 An integer vector preserving all answers

Lemma 12. *Let $M \in \{0,1\}^{k \times n}$ and let $R \geq 1$ be an integer. If*

$$(R+1)^n > (nR+1)^k,$$

then there is a nonzero $z \in \mathbb{Z}^n$ such that $Mz = 0$ and $|z_i| \leq R$ for every $i \in [n]$.

Proof. There are $(R+1)^n$ vectors $x \in \{0,1,\dots,R\}^n$. Each coordinate of Mx is a sum of at most n coordinates of x , since every entry of M is either 0 or 1. Each summand is an integer between 0 and R , so each coordinate of Mx is an integer between 0 and nR . Each of the k coordinates of Mx has at most $nR+1$ possible values, namely $0, 1, \dots, nR$. Hence, as x ranges over $\{0,1,\dots,R\}^n$, the vector Mx takes at most $(nR+1)^k$ distinct values. By our assumption and the pigeonhole principle, there exist two distinct vectors $x, y \in \{0,1,\dots,R\}^n$ with the same image. Set $z = x - y$. Since $x \neq y$, we have $z \neq 0$. Moreover, $Mz = Mx - My = 0$, and $|z_i| \leq R$ for every $i \in [n]$. $\square$

We now combine the preceding two lemmas to prove the lower bound. We answer queries according to a regular tournament and show that, after too few queries, there is a realizable degree vector consistent with all answers in which the algorithm's output does not have maximum outdegree.

Theorem 13. *Let $n \geq 17$ be odd. Then*

$$D_{\text{dc}}(\text{KING}_n) \geq \frac{n}{2} - \frac{3n}{2 \log n} - 1.$$

Proof. For this proof, fix $m = (n-1)/2$ and $R = \lfloor n/8 \rfloor$. Fix a correct deterministic algorithm and run it on an n -vertex regular tournament, that is, one in which every vertex has outdegree m . It is well known that such a tournament exists for every odd n [Moo68].

Suppose this execution makes q queries $S_1, \dots, S_q$ and outputs vertex v . Define $M \in \{0,1\}^{(q+1) \times n}$ by

$$M_{1,j} = 1, \quad M_{i+1,j} = \begin{cases} 1 & j \in S_i, \\ 0 & j \notin S_i, \end{cases} \quad i \in [q], j \in [n].$$

Assume, towards a contradiction, that

$$(R+1)^n > (nR+1)^{q+1}. \tag{4}$$

By Lemma 12, there is a nonzero integer vector z with $Mz = 0$ and $|z_i| \leq R$. The first row gives $\sum_i z_i = 0$, and the remaining rows give

$$\sum_{j \in S_i} z_j = 0 \quad (i \in [q]). \quad (5)$$

In particular, z is not constant. We may replace z by $-z$ so that z_v is not a maximum coordinate of z : if z_v is already nonmaximum, keep z ; otherwise choose w with $z_w < z_v$ and negate z .

Set $d_i = m + z_i = (n-1)/2 + z_i$ for every $i \in [n]$. Then $\mathbf{d} = (d_1, \dots, d_n)$ is an integer vector with total sum $n(n-1)/2 = \binom{n}{2}$. Moreover,

$$m - R \geq \frac{3n}{8} - \frac{1}{2} \geq \frac{n}{4}, \quad m + R \leq \frac{5n}{8} - \frac{1}{2} \leq \frac{3n}{4}.$$

Thus, each $d_i \in [n/4, 3n/4]$. By Lemma 11, $\mathbf{d}$ is realizable. Choose w with $z_w > z_v$. By Lemma 6, there is a realization T' of $\mathbf{d}$ in which w covers v .

For each queried set S_i , the sum of the outdegrees in T' is

$$\sum_{j \in S_i} d_j = \sum_{j \in S_i} (m + z_j) = m|S_i| + \sum_{j \in S_i} z_j = m|S_i|,$$

where the last equality follows from Equation (5). This is the same degree sum as in the regular tournament. By Equation (1), the answer to the query S_i is therefore the same on both tournaments. This holds true for all $i \in [q]$. The algorithm therefore follows the same execution on T' and outputs v . This is incorrect because w covers v in T' and hence v cannot be a maximum-outdegree vertex or king in T' by Observation 4.

Thus Equation (4) is impossible. Taking logs and cross-multiplying gives

$$q + 1 \geq \frac{n \log(R+1)}{\log(nR+1)}.$$

Since $R = \lfloor n/8 \rfloor$, we have $R+1 \geq n/8$ and $nR+1 \leq n^2$. Therefore,

$$q + 1 \geq \frac{n \log(R+1)}{\log(nR+1)} \geq \frac{n(\log n - 3)}{2 \log n} = \frac{n}{2} - \frac{3n}{2 \log n}.$$

Thus $q \geq n/2 - 3n/(2 \log n) - 1$, proving the claimed lower bound. $\square$

3.3 Proof of main result

Lemma 14. *For every $n \geq 2$, $D_{\text{dc}}(\text{KING}_n) \geq D_{\text{dc}}(\text{KING}_{n-1})$.*

Proof. Given a tournament T on $[n-1]$, add a new vertex s and direct every edge incident to s towards s , obtaining T' . For a query $S \subseteq [n-1] \cup \{s\}$, put $A = S \cap [n-1]$. Then

$$c_{T'}(S) = c_T(A) + \begin{cases} |A| & s \notin S, \\ 0 & s \in S. \end{cases}$$

Thus one query to T suffices to simulate a query to T' . The vertex s cannot be a king of T' , and it cannot be an internal vertex of a directed path. Consequently, every king of T' is a king of T . An algorithm for KING_n therefore yields one for KING_{n-1} with no more queries. $\square$

Proof of Theorem 1. The upper bound is Proposition 9. Let m be the largest odd integer at most n . By Lemma 14 and Theorem 13, for all sufficiently large n ,

$$D_{\text{dc}}(\text{KING}_n) \geq D_{\text{dc}}(\text{KING}_m) \geq \frac{m}{2} - \frac{3m}{2 \log m} - 1.$$

Since $m \geq n-1$, the right-hand side is at least $(1/2 - \epsilon)n$ for every fixed $\epsilon > 0$ and all sufficiently large n . In particular, it exceeds $n/3$ for all sufficiently large n . $\square$

4 An alternative lower bound via communication

The structural equivalence in Corollary 8 also allows us to deduce an $\Omega(n)$ lower bound from the deterministic maximum-outdegree communication lower bound of $\Omega(n \log n)$ of Mande et al. [MPSS24, Theorem 5].

Fix a publicly known partition of the unordered vertex pairs between Alice and Bob. Each player knows the directions of the edges assigned to them. We now show how Alice and Bob can simulate a q -cost cut-query algorithm by an $O(q \log n)$ -cost communication protocol. To simulate a directed cut query S , each player counts the edges they own directed from S to its complement and sends this count to the other player. Each count is at most $\binom{n}{2}$, so this can be sent to each other using $O(\log n)$ bits of communication. Both players then know the answer to the cut-query S and can determine the next query of the deterministic cut-query algorithm without further communication. Since the cut-query algorithm makes at most q queries, the cost of the communication protocol is $O(q \log n)$.

By Corollary 8, a correct cut-query algorithm for finding a king must output a maximum-outdegree vertex on every tournament. The $\Omega(n \log n)$ communication lower bound for a suitable partition therefore implies $q = \Omega(n)$.

5 Further directions

An interesting special case is finding a king when the input is promised to be a transitive tournament. Since its outdegrees form a permutation of $\{0, \dots, n-1\}$ and its unique king is its source, Equation (1) makes this equivalent to finding the location of n in an unknown permutation of $[n]$, where each query returns the sum of an arbitrary subset of entries. Determining the query complexity of this problem seems to be an independently interesting puzzle.

Another natural next question is to determine the *randomized* directed cut-query complexity of finding a king. Let $R_{\text{dc}}(\text{KING}_n)$ denote the minimum worst-case query bound of an algorithm that outputs a king with probability at least $2/3$ on every input. The deterministic algorithm from Proposition 9 gives $R_{\text{dc}}(\text{KING}_n) \leq n-1$. The randomized $\Omega(n)$ communication lower bound for finding a king [MPSS24, Theorem 4], together with the simulation in Section 4 (which can also easily be seen to work in the randomized setting), gives

$$\Omega\left(\frac{n}{\log n}\right) \leq R_{\text{dc}}(\text{KING}_n) \leq n-1.$$

Does randomization permit an $o(n)$ -cost cut-query algorithm to find a king, or does the linear lower bound persist?

Our structural equivalence between KING_n and MOD_n in Corollary 8 relies on correctness on every input. For each nonmaximum vertex, Lemma 6 gives a realization in which that vertex is not a king, but this realization may depend on the vertex. Moreover, the tournament constructed in the proof of Theorem 13 depends on the queries made by the algorithm. These arguments therefore do not directly yield a matching randomized lower bound via Yao’s principle, which requires a single hard input distribution against all deterministic algorithms with a given query budget.

References

- [AEG⁺22] Simon Apers, Yuval Efron, Paweł Gawrychowski, Troy Lee, Sagnik Mukhopadhyay, and Danupon Nanongkai. Cut query algorithms with star contraction. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 507–518. IEEE, 2022.
- [AGNS24] Amir Abboud, Tomer Grossman, Moni Naor, and Tomer Solomon. From donkeys to kings in tournaments. In *32nd Annual European Symposium on Algorithms, ESA 2024*, volume 308 of *LIPIcs*, pages 3:1–3:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

- [ASW25] Aditya Anand, Thatchaphol Saranurak, and Yunfan Wang. Deterministic edge connectivity and max flow using subquadratic cut queries. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 124–142. SIAM, 2025.
- [BJRS22] Arindam Biswas, Varunkumar Jayapaul, Venkatesh Raman, and Srinivasa Rao Satti. Finding kings in tournaments. *Discrete Applied Mathematics*, 322:240–252, 2022.
- [CL23] Deeparnab Chakrabarty and Hang Liao. A query algorithm for learning a spanning forest in weighted undirected graphs. In *Proceedings of the 34th International Conference on Algorithmic Learning Theory*, volume 201 of *Proceedings of Machine Learning Research*, pages 259–274. PMLR, 2023.
- [GKMM26] Prantar Ghosh, Sahil Kuchlous, Shravan Mehra, and Sagnik Mukhopadhyay. The power of the score sequence of a tournament. In *34th Annual European Symposium on Algorithms (ESA 2026)*, volume 388 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 156:1–156:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026.
- [IM25] Ziad Ismaili Alaoui and Nikhil S. Mande. Hardness of finding kings and strong kings. In *45th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS*, volume 360 of *LIPIcs*, pages 36:1–36:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025.
- [KM26] Yotam Kenneth-Mordoch. Deterministic global minimum cut using $\tilde{O}(n^{3/2})$ cut queries. *arXiv preprint arXiv:2610.23461*, 2026.
- [Lan53] H. G. Landau. On dominance relations and the structure of animal societies: III. the condition for a score structure. *The Bulletin of Mathematical Biophysics*, 15(2):143–148, 1953.
- [LRT22] Oded Lachish, Felix Reidl, and Chhaya Trehan. When you come at the king you best not miss. In *42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2022*, volume 250 of *LIPIcs*, pages 25:1–25:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
- [MN20] Sagnik Mukhopadhyay and Danupon Nanongkai. Weighted min-cut: sequential, cut-query, and streaming algorithms. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 496–509, 2020.
- [Moo68] John W Moon. Topics on tournaments. *New York*, 1968.
- [MPS23] Nikhil S. Mande, Manaswi Paraashar, and Nitin Saurabh. Randomized and quantum query complexities of finding a king in a tournament. In *43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS*, volume 284 of *LIPIcs*, pages 30:1–30:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.
- [MPSS24] Nikhil S. Mande, Manaswi Paraashar, Swagato Sanyal, and Nitin Saurabh. On the communication complexity of finding a king in a tournament. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2024)*, volume 317 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 64:1–64:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024.
- [RSW18] Aviad Rubinfeld, Tselil Schramm, and S. Matthew Weinberg. Computing exact minimum cuts without knowing the graph. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*, volume 94 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 39:1–39:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018.
- [SSW03] Jian Shen, Li Sheng, and Jie Wu. Searching for sorted sequences of kings in tournaments. *SIAM Journal on Computing*, 32(5):1201–1209, 2003.