Sorting and Selection Problems in Partially Ordered Sets

Merkouris Papamichail

Master's Program in Algorithms, Logic, and Discrete Mathematics
National and Kapodistrian University of Athens,
National Technical University of Athens

September 2022, Athens

Talk Outline

Talk Outline

Traditional Problems

In the traditional sorting setting:

1. We examine *total orders*.
2. We use time complexity, as a measure of efficiency.
3. We can compare *any* two elements *directly*.
4. For the Sorting problem (Comparison-Based Model):
    1. Lower Bound: $\Omega(n \log n)$.
    2. Upper Bound: $O(n \log n)$ (e.g. Merge Sort)
5. For the k-Selection problem (Comparison-Based Model):
    1. Lower Bound: $n - k \log \left(\binom{n}{k-1} / k \right)$ [10].
    2. Upper Bound: $O(n)$ (e.g. Median of Medians [3])

Our Approach

In this thesis:

1. We generalize the setting to examine *partially ordered sets*.
2. We use the *query complexity* as measure of efficiency.
3. We can compare two elements only by *querying an oracle function*.
4. We examine different query-based models.
   1. Width-Based Model, Daskalakis et al. [5].
   2. Forbidden Comparisons Model, Banerjee and Richards [1, 2].

Width-Based Model (WBM)

Width-Based Model (WBM)

1. A finite set $\mathcal{U}$.
2. An oracle function $c\colon \mathcal{U} \times \mathcal{U} \to \{\preccurlyeq, \succ, \not\sim\}$, respecting an underlying, unknown partial order $\mathcal{R}_{\preccurlyeq}$.
3. An *upper bound* w to the poset's *width*.

- Introduced by Daskalakis et al. [5], 2011.

Forbidden Comparisons Model (FCM)

Forbidden Comparisons Model (FCM)

1. A finite set $\mathcal{U}$.
2. An oracle function $c\colon \mathcal{U} \times \mathcal{U} \to \{\preccurlyeq, \succ, \perp\}$, respecting an underlying, unknown partial order $\mathcal{R}_{\preccurlyeq}$.
   - If $c(a, b) = \perp$, then *we are not allowed to compare* a, b.
   - We (may be able to) deduce their relation by *transitivity*.
3. A *comparison graph* $G = (V, E)$.
   - If $\{a, b\} \in E$, then a, b are *incomparable*, i.e. $c(a, b) \neq \perp$
4. The *number of missing edges* q.
   - $q = \binom{|V|}{2} - |E|$.

▶ Introduced by Banerjee and Richards [1], 2016. Improved upon by Biswas et al. [2] 2017.

Thesis Organization

Ch. 1: Introduction

Ch. 2: Network Flows & Posets:

- Flows & Networks
- Matchings & Bipartite Graphs
- Minimal Chain Decomposition

Ch. 3: WBM: Sorting

- IT Lower Bound $O(n(\log n + w))$
- ChainMerge D.S.
- Bin-Insertion Sort $O(wn \log n)$
- Entropy Sort $O(n(\log n + w))$
- Merge Sort $O(wn)$

Ch. 4: WBM: Selection

- Selection (Det. & Rand.)
- k-Selection (Det. & Rand.)
- Det. Selection Lower Bound (Adv.)
- Det. k-Selection Lower Bound (Adv.)
- Rand. k-Selection Lower Bound

Ch. 5: FCM: Sorting

- Banerjee & Richards Alg. $O((n+q)\log(n))$
- Biswas et al. Alg. $O((n+q)\log(n^2/q))$
- Lower Bound $O(q + n \log n)$
- Special Cases $O(n \log n)$
  - Chordal Graphs
  - Comparability graphs

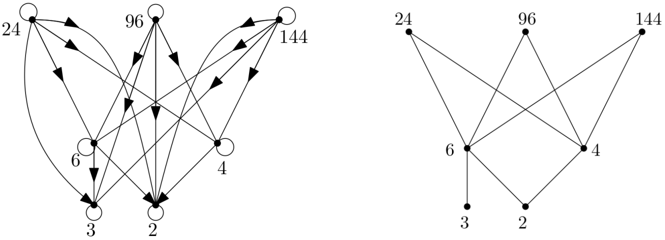
Ch. 6: Conclusions

Talk Outline

Definition & Representation

Partially Ordered Sets

Let $\mathcal{U}$ be a finite set. Also let $\mathcal{R}_{\preccurlyeq} \subseteq \mathcal{U} \times \mathcal{U}$. We call the pair $\langle \mathcal{U}, \mathcal{R}_{\preccurlyeq} \rangle$ a *partially ordered set*, or *poset*, if $\mathcal{R}_{\preccurlyeq}$ is *reflexive*, *antisymmetric*, and *transitive*. Additionally, if for every two elements $x, y \in \mathcal{U}$ are related, i.e., $x \sim y$; we call $\langle \mathcal{U}, \mathcal{R}_{\preccurlyeq} \rangle$ a *total order*.



(a) Directed graph $D = (\mathcal{U}, \mathcal{R}_{\preccurlyeq})$.

(b) Hasse diagram.

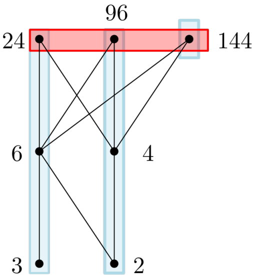
Figure 1: Representing a poset. The above partial order is the "divides" relation.

Chains & Antichains

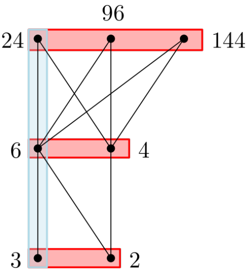
Let $\mathcal{V} \subseteq \mathcal{U}$ be a subset of the partial order $\langle \mathcal{U}, \mathcal{R}_{\preccurlyeq} \rangle$, then:

- If $\langle \mathcal{V}, \mathcal{R}_{\preccurlyeq} \rangle$ is a *total order*, then we call $\mathcal{V}$ a *chain*.
- If every $a, b \in \mathcal{V}$ are *unrelated* $a \not\sim b$, then we call $\mathcal{V}$ a *antichain*.

- If $\mathcal{C}$ a *maximum chain*, we call the size of $\mathcal{C}$ the *height* of the poset.
- If $\mathcal{A}$ a *maximum antichain*, we call the size of $\mathcal{A}$ the *width* of the poset.



(a) Dilworth's theorem. The number of chains in a minimal decomposition equals the size of a maximum antichain.



(b) Mirsky's theorem. The number of antichains in a minimal decomposition equals the size of a maximum chain.

Figure 2: Dilworth's theorem (1950) and its dual due to Mirsky (1971).

The Problems

- Let $\langle \mathcal{U}, \mathcal{R}_{\preccurlyeq} \rangle$ be a poset. For some element $x \in \mathcal{U}$ we call *height* of x, and write height(x), the size of the maximum chain $\mathcal{V}$, such that for each $v \in \mathcal{V}$, $x \succcurlyeq v$.

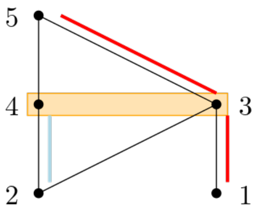
The problems we examine are the following:

1. Sorting: Find the underlying, *unknown* relation $\mathcal{R}_{\preccurlyeq}$.
2. k-Selection: Find the *k-smallest elements*, i.e., the elements of height k.

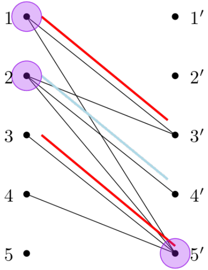
- We want to solve the above problems *with as few queries as possible*.

Finding a Minimal Chain Decomposition

1. From a poset $\langle \mathcal{U}, \mathcal{R}_{\preccurlyeq} \rangle$.
2. Create the corresponding bipartite graph $G = (\mathcal{U} \uplus \mathcal{U}', \mathcal{R}'_{\preccurlyeq})$.
3. Find a *maximum matching* M in G.
4. "Glue" the edges of the matching together to create the decomposition.



(a) A poset on five elements.

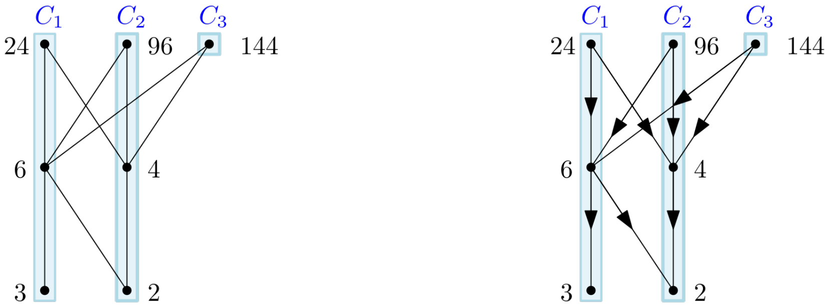


(b) The corresponding bipartite graph of the poset on the left.

Figure 3: A poset on five elements and the corresponding bipartite graph.

Talk Outline

ChainMerge Data Structure



(a) The Hasse diagram.

(b) The corresponding ChainMerge data structure.

Figure 4: The Hasse diagram and the corresponding ChainMerge data structure, represented as a directed graph. The arcs represent an index.

Construction: $O(wn)$ query and time complexity.
Answering Queries: $O(1)$ time, with no additional queries to oracle.

Information-Theoretic Lower Bound

Theorem: Brightwell and Goodfall [4], 1996

Let $N_w(n)$ be the number of partially ordered sets, of width at most w, on n elements. We have,

$$\frac{n!}{w!}4^{n(w-1)}n^{-24w(w-1)} \leq N_w(n) \leq n!4^{n(w-1)}n^{-(w-2)(w-1)/2}w^{w(w-1)/2}$$

Theorem: Lower Bound, Daskalakis et al. [5], 2011

Any algorithm which sorts a poset of width at most w on n elements requires $\Omega(n(\log n + w))$ queries.

The Algorithm Bin-Insertion Sort

1. We keep a chain decomposition $\mathcal{C} = \{C_1, \ldots, C_w\}$
2. While there are unsorted elements:
    1. Let e be an *as yet unsorted* element.
    2. For each chain $C_i \in \mathcal{C}$:
        1. Do *binary search* to find the *greatest element* of C_i that is *dominated by* e.
        2. Do *binary search* to find the *smallest element* of C_i that *dominates* e.
    3. *Recompute* a *minimal chain decomposition* $\mathcal{C}'$.
3. From the chain decomposition $\mathcal{C}$, we compute the ChainMerge Data Structure.
4. Return the ChainMerge Data Structure.

Bin-Insertion Sort: Conclusions

Theorem: Faigle and Túran [9], 1985

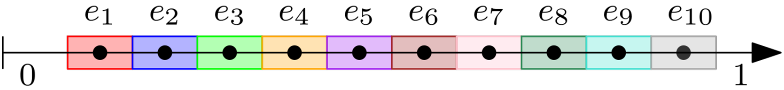
Bin-Insertion Sort sorts any partial order $\langle \mathcal{U}, \mathcal{R}_{\preccurlyeq} \rangle$ of width at most w on n elements with $O(wn \log n)$ oracle queries.

Remarks:

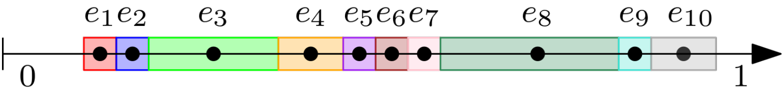
- The binary searches take no advantage of the *structural properties* of the poset.
- Uninformative queries cost us the same as informative ones.
- We patch the Bin-Insertion Sort, in order to correct the above issues.
- We will *amortize* the cost of queries over the insertions.

How are we are going to achieve this?

Weighted Binary Search: A Visual Representation



(a) Equal intervals for each element.



(b) Intervals corresponding to the weight of each element.

Figure 5: A visualization of the Weighted Binary Search.

Uninformative queries will correspond to large *mass* (weight).

Weighted Binary Search: Poset Extensions I

- Let e be the element we would like to sort.
- Let C_i be the i-th chain.
- $C_i = \{e_{i_1}, e_{i_2}, \ldots, e_{i_\ell}\}$.

▶ We want to describe the "*universes*" where, e_{i_k} *is the smallest element of* C_i *that dominates* e.

▶ We compile the following lists of restrictions:

1. All the elements greater that e_{i_k}, will *also* dominate e:
   - $\mathcal{ER}_k = \{e \preccurlyeq e_{i_j} \mid j \in \{k, k+1, \ldots, \ell\}\}$ (*Enforced Relations*)
2. All the elements smaller than e_{i_k} *cannot* dominate e:
   - $\mathcal{PR}_k = \{e \preccurlyeq e_{i_j} \mid j \in \{1, 2, \ldots, k-1\}\}$ (*Prohibited Relations*)

Weighted Binary Search: Poset Extensions II

- Let $P' = \langle \mathcal{U}', \mathcal{R}'_{\preccurlyeq} \rangle$ be the *already computed* poset, before we sort e.
  - Let $\mathcal{U}'' = \mathcal{U}' \cup e$.
  - Let $\mathcal{R}''_{\preccurlyeq} \supseteq \mathcal{R}'_{\preccurlyeq}$, be a new set of relations, *containing* e, where:
    1. $\mathcal{R}''_{\preccurlyeq} \cap \mathcal{ER}_k = \mathcal{ER}_k$, i.e., *respects all the enforced relations.*
    2. $\mathcal{R}''_{\preccurlyeq} \cap \mathcal{PR}_k = \varnothing$, i.e., *avoids all prohibited relations.*
  - We call $P'' = \langle \mathcal{U}'', \mathcal{R}''_{\preccurlyeq} \rangle$ a *w-width extension* *of P' conditioned on* $(\mathcal{ER}_k, \mathcal{PR}_k)$.

P'' describes *a possible "universe"* where e_{i_k} is the smallest element that dominates e.

Weighted Binary Search: Mass Function

1. Let D_{i_k} be the *number of w-width extensions conditioned on* $(\mathcal{ER}_k, \mathcal{PR}_k)$.
2. D_{i_k} denotes the number of universes where e_{i_k} is the smallest element of the i-th chain that dominates e (the element we want to sort).
3. Let $D_i = \sum_{k=1}^{\ell_i+1} D_{i_k}$, the number of *all possible* universes.

▶ We define the mass function of the elements of the i-th chain, for e being the new element, as,

$$\mathtt{mass}_{i,e}(e_{i_k}) = \frac{D_{i_k}}{D_i}$$

▶ The k-th interval of the weighted binary search, will have size equal to $\mathtt{mass}_{i,e}(e_{i_k})$.

Summary

We call the above "patched" version of Bin-Insertion Sort, using weighted binary searches Entropy Sort [5].

Summary

- Entropy Sort is query-optimal. Query Complexity: $O(n \log n + nw)$.
- The computation of the masses needs *exponential time*.
- Entropy Sort is time-inefficient.
- In [5], the authors the Merge Sort algorithm, as a time-efficient alternative, that *is not* query optimal.
- Merge Sort: Query Complexity $O(wn \log(n/w))$, Time Complexity $O(w^2 n \log(n/w))$

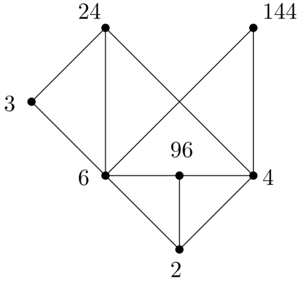
Talk Outline

Forbidden Comparisons Model: Revision

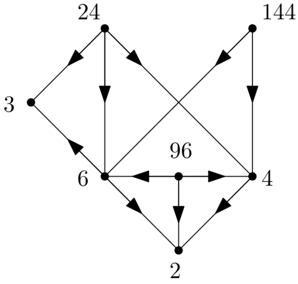
- Remember: In the Forbidden Comparison Model:
  1. We are given an *undirected comparison graph* $G = (V, E)$.
  2. We are *only* allowed to compare edge-connected elements.
  3. We know that edge-connected elements *are related*.
  4. q denotes *the number of missing edges* from G.

Forbidden Comparisons Model: Strategy

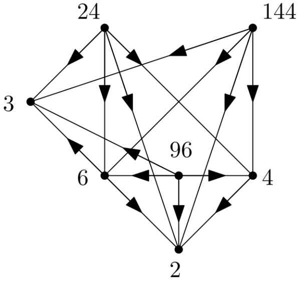
- Our Strategy:
  1. Find the orientation of the edges of G.
  2. Compute the transitive edges.



(a) Comparison graph.



(b) Assigning orientation to the comparison graph.



(c) Computing the transitive edges.

Forbidden Comparisons Model: Main Idea

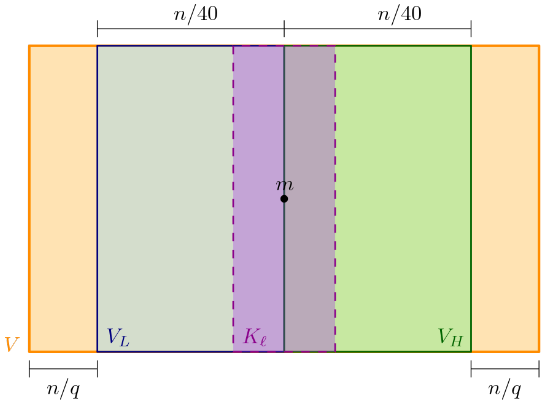
- Key Property: Let K_ℓ be a *subgraph* of the comparison graph $G = (V, E)$, then the elements of $V(K_\ell)$ are *totally ordered*.

Main Idea

1. Find a big enough clique K_ℓ in G, using a greedy strategy.
2. Find the median m of $V(K_\ell)$.
3. Use m as *"approximate" median* of V.
4. Separate V to V_L and V_H:
    1. V_L contains elements less than or equal to m.
    2. V_H contains elements greater than m
5. Recurse through the smaller sets V_L, V_H.

Key Lemma, (Banerjee and Richards [1], 2016)

If $q < n^2/320$, there is an approximate *median* m, that is greater than at least $n/40$ elements and less than $n/40$ elements. Furthermore m cannot be compared with at most $O(q/n)$ elements, and it can be found using $O(q+n)$ queries.



Forbidden Comparisons Model: Special Cases

In chordal and comparability graphs:

1. Sufficient to "locally" sort an element, with respect to its neighbors.
2. We can assume that their neighbors are *totally ordered*.
3. We do a binary search, with respect to its neighbors.
4. Our output is a *topological order* of the nodes.

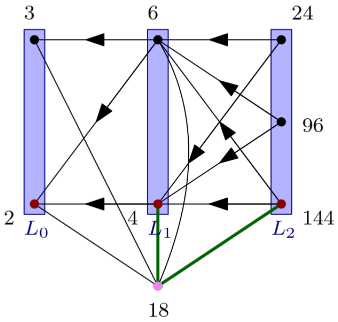


Figure 7: Comparability Graphs: Adding the vertex $v = 18$ to our running Example.

Summary

- Original Algorithm [1] has query complexity $O((q+n)\log n)$.
- Improvement due to [2] has query complexity $O((q+n)\log(n^2/q))$
- Lower Bound $O(q+n\log n)$.
- In [2], the authors explore the structural properties of chordal and comparability graphs.
- In these graph families, it suffices to compute a *topological order*. Query complexity $O(n\log n)$.

Talk Outline

Conclusions

In this presentation..

1. We examined the Sorting problem in Posets.
2. We considered two Query-Based models:
    - Width-Based Model (WBM)
    - Forbidden Comparisons Model (FCM)
3. In Width-Based Model (Sorting):
    - We established an Information Theoretic Lower Bound.
    - We presented a naive sorting algorithm (Bin-Insertion Sort).
    - We discussed a query-optimal algorithm (Entropy Sort).
4. In Forbidden Comparisons Model (Sorting):
    - We presented an improved sorting algorithm, with $O((q+n)\log(n^2/q))$ query complexity.
    - We discussed some special cases with $O(n\log n)$ query complexity.

Future Work

The main Open Problems are the following:

1. Width-Based Model (Sorting):
    - Achieve query-optimality in polynomial time.
    - The authors suggested volume approximation techniques as a possible avenue of improvement [8].
2. Width-Based Model (Selection):
    - Close the gap between upper and lower bounds.
3. Forbidden Comparisons Model:
    - Close the gap between upper and lower bounds, in Selection.
    - What's the complexity of Sorting, when the underlying partial order, is a total order?
    - Explore the k-Selection problem in this setting.

Thank you for your time!

I'll be happy to answer your questions :)

Talk Outline

Width-Based Model: Sorting

	Query Complexity	**Time Complexity**	**Lower Bound**
Bin-Insertion Sort	$O(wn \log n)$	$O(n^4 - n^2 w^2 + w \log n)$	
Entropy Sort	$O(n \log n + wn)$	Exponential	$\Omega(n(\log n + w))$
Merge Sort	$O(wn \log(n/w))$	$O(w^2 n \log(n/w))$	

Table 1: The Algorithms for Sorting in Widrth-Based Model.

Width-Based Model: Selection

	Query Complexity	Time Complexity	Lower Bound
Det. Selection	$O(wn)$	$O(wn)$	$\Omega\left(\frac{w+1}{2}n - w\right)$
Random. Selection	$\frac{w+1}{2}n + \frac{w^2-w}{2}(\log n - \log w)$	$\frac{w+1}{2}n + \frac{w^2-w}{2}(\log n - \log w)$	$\Omega\left(\frac{w+3}{4}n - w\right)$
Det. k-Selection (Entropy Sort)	$O(16wn + 4n\log(2k) + 6n\log w)$	Exponential	(4.1)
Det. k-Selection (Merge Sort)	$O(8wn\log(2k))$	$O(w^2 n\log(2k))$	(4.1)
Random. k-Selection	$O(wn + 16kw^2\log(2k))$	$O(wn + \mathrm{poly}(k,w)\log n)$	(4.2)

Table 2: The Algorithms for Selection in the Width-Based Model.

Forbidden Comparisons Model: Sorting

	Query Complexity	**Time Complexity**	**Lower Bound**
Sorting [1]	$O((q+n)\log n)$	$O(n^2 + q^{\omega/2})$	$\Omega(q + n\log n)$
Sorting [2]	$O((q+n)\log(n^2/q))$	$O(n^2 + q^{\omega/2})$	$\Omega(q + n\log n)$
Sorting (Chordal Graphs)	$O(n\log n)$	$O(n^{\omega})$	$O(n\log n)$
Sorting (Comparability Graphs)	$O(n\log n)$	$O(n^{\omega})$	$O(n\log n)$
Random Sort	$\widetilde{O}(n^2\sqrt{q+n} + n\sqrt{q})$	$O(n^{\omega})$	Open Question

Table 3: The Algorithms of Sorting in the Forbidden Comparisons Model.

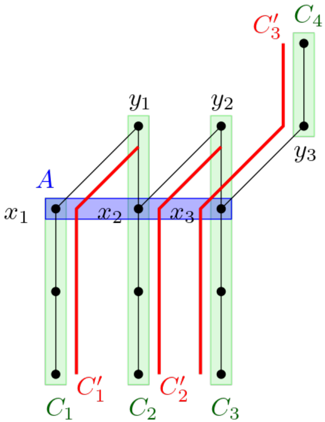


Figure 8: Peeling Example. We reduce the chain decomposition from $\{C_1, C_2, C_3, C_4\}$ to $\{C_1', C_2', C_3'\}$. With A we denote a maximum chain. The pairs $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ form a dislodgement sequence.

Posets: Order Dimension

▶ Let $P = \langle \mathcal{U}, \mathcal{R}_{\preccurlyeq} \rangle$ be a poset. We call $L = \langle \mathcal{U}, \mathcal{R}'_{\preccurlyeq} \rangle$ a *linear extension* if $\mathcal{R}'_{\preccurlyeq} \supseteq \mathcal{R}_{\preccurlyeq}$ and L is a *total order*.

Definition: Order Dimension [7]

The *order dimension* δ of a poset $P = \langle \mathcal{U}, \mathcal{R}_{\preccurlyeq} \rangle$ is the smallest number $\delta \in \mathbb{N}$, such that $\mathcal{L} = \{L_1, \ldots, L_\delta\}$ is a *family of linear extensions* of P, where,

$$P = \bigcap_{i \in [\delta]} L_i$$

- Finding the order dimension of a partial order is *NP-complete* [13].
- If w is the width of a poset, we have $\delta \leq w$ [6].

Dominance Drawing Dimension

Let $P = \langle \mathcal{U}, \mathcal{R}_{\preccurlyeq} \rangle$ be a poset. Let $\phi \colon \mathcal{U} \to \mathbb{R}^d$ a a mapping from the partially ordered set to the Euclidean d-dimensional space, where,

$$x \preccurlyeq y \Leftrightarrow \phi(x) \leq \phi(y).$$

We call the smallest dimension that the above is possible, *drawing dominance dimension*.

- *It turns out* $\delta = d$ [11].

Notes:

- This way we could *traverse* from discrete mathematics to geometry.
- Hence, we could use the volume approximation techniques (as suggested in [5]).
- Given the dominance drawing dimension δ, we could create a dominance drawing in *polynomial time* $O(\delta n)$ [12].

Bibliography I

I. Banerjee and D. S. Richards, *Sorting under forbidden comparisons*, in 15th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2016, June 22-24, 2016, Reykjavik, Iceland, R. Pagh, ed., vol. 53 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016, pp. 22:1–22:13.

A. Biswas, V. Jayapaul, and V. Raman, *Improved bounds for poset sorting in the forbidden-comparison regime*, in Algorithms and Discrete Applied Mathematics - Third International Conference, CALDAM 2017, Sancoale, Goa, India, February 16-18, 2017, Proceedings, D. R. Gaur and N. S. Narayanaswamy, eds., vol. 10156 of Lecture Notes in Computer Science, Springer, 2017, pp. 50–59.

M. Blum, R. W. Floyd, V. R. Pratt, R. L. Rivest, and R. E. Tarjan, *Time bounds for selection*, J. Comput. Syst. Sci., 7 (1973), pp. 448–461.

G. R. Brightwell and S. J. Goodall, *The number of partial orders of fixed width*, Order, 20 (2003), pp. 333–345.

C. Daskalakis, R. M. Karp, E. Mossel, S. J. Riesenfeld, and E. Verbin, *Sorting and selection in posets*, SIAM J. Comput., 40 (2011), pp. 597–622.

R. P. Dilworth, *A decomposition theorem for partially ordered sets*, Annals of Mathematics, 51 (1950), pp. 161–166.

Bibliography II

B. Dushnik and E. W. Miller, *Partially ordered sets*, American Journal of Mathematics, 63 (1941), pp. 600–610.

M. E. Dyer, A. M. Frieze, and R. Kannan, *A random polynomial time algorithm for approximating the volume of convex bodies*, in Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA, D. S. Johnson, ed., ACM, 1989, pp. 375–381.

U. Faigle and G. Turán, *Sorting and recognition problems for ordered sets*, in STACS 85, 2nd Symposium of Theoretical Aspects of Computer Science, Saarbrücken, Germany, January 3-5, 1985, Proceedings, K. Mehlhorn, ed., vol. 182 of Lecture Notes in Computer Science, Springer, 1985, pp. 109–118.

F. Fussenegger and H. N. Gabow, *A counting approach to lower bounds for selection problems*, J. ACM, 26 (1979), pp. 227–238.

E. Milner and M. Pouzet, *A note on the dimension of a poset*, Order, 7 (1990), pp. 101–102.

G. Ortali and I. G. Tollis, *Multidimensional dominance drawings*, CoRR, abs/1906.09224 (2019).

M. Yannakakis, *The complexity of the partial order dimension problem*, SIAM Journal on Algebraic Discrete Methods, 3 (1982), pp. 351–358.