

Searching, Sorting and Randomised
Algorithms for Central Elements and
Ideal Counting in Posets

D. Dubhashi K. Mehlhorn D. Ranjan
Ch. Thiel

MPI-I-93-154

November 1993

Searching, Sorting and Randomised Algorithms for Central Elements and Ideal Counting in Posets

Devdatt P. Dubhashi¹, Kurt Mehlhorn², Desh Ranjan^{1,3} and Christian Thiel¹

¹ Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany

² Max-Planck-Institut für Informatik and Fachbereich Informatik, Universität des Saarlandes, Im Stadtwald, D-66123 Saarbrücken, Germany

³ Dept. of Computer Science, New Mexico State University, Las Cruces, USA

Abstract. By the Central Element Theorem of Linial and Saks, it follows that for the problem of (generalised) searching in posets, the information-theoretic lower bound of $\log N$ comparisons (where N is the number of order-ideals in the poset) is tight asymptotically. We observe that this implies that the problem of (generalised) sorting in posets has complexity $\Theta(n \cdot \log N)$ (where n is the number of elements in the poset). We present schemes for (efficiently) transforming a randomised generation procedure for central elements (which often exists for some classes of posets) into randomised procedures for approximately counting ideals in the poset and for testing if an arbitrary element is central.

1 Introduction

In this paper we are concerned with the general problem of searching and sorting ordered data structures and some related issues and problems that arise in this context. In the general situation, we are given a partially-ordered data structure modelled by a poset P , and a *storage function* $f : P \rightarrow \mathbb{R}$ such that $p \leq_P q$ implies $f(p) \leq f(q)$. We may think of the elements of the poset as data locations where real numbers are stored consistent with the poset ordering.

In the *searching* problem, first considered by Linial and Saks, [5], we are given a real number α and asked to locate it in the poset (if it is present) with a minimum number of comparisons of the form $\alpha \leq f(p)$ or $\alpha = f(p)$, for $p \in P$. They showed that the direct information-theoretic lower bound of $\log N$ (where N is the number of *ideals* in P) can indeed be attained *if* one can locate so-called *central* elements (section 2) in the poset. Hence for the problem of searching ordered data structures, the search for central elements assumes central importance. They went on to prove a deep theorem which asserted that *every* poset in fact possesses such central elements. Their proof was, however, non-constructive and the question of whether such elements can actually be found constructively was left open.

The *sorting* problem for general posets is: given a finite set P and an *unknown* partial order \leq on P , determine the partial order using a minimum number of comparisons of the form $p \leq q$ for $p, q \in P$. The central elements

again play a key role. Given a procedure to obtain central elements, the sorting problem for posets with n elements and having a total of N ideals can be solved with $O(n \log N)$ comparisons. Moreover, we demonstrate an information-theoretic lower-bound of $\Omega(n \log N)$ for this problem.

Faigle *et al* show an intimate connection between the problem of generating central elements and the problem of counting ideals in the poset [2]. The latter problem is known to be $\#P$ -complete in general [6]. We strengthen this connection by extending it in a probabilistic setting. One of our motivations is to investigate if randomisation helps in solving the approximate version of this $\#P$ -complete problem as it indeed does in some other well known cases, [4, 1]. We offer some progress in this direction although the complete solution still appears distant. In particular, we prove that it is possible to amplify the correctness probability of a probabilistic procedure that outputs central elements with probability greater than $1/2$.

The paper is organised as follows. Section 2 introduces some basic definitions and concepts. In section 3, we consider the problem of generalised sorting in posets and give matching upper and lower bounds for the problem (at least non-constructively). In section 4, we pursue the intimate relation between the problem of generating central elements and the problem of counting ideals in posets, first observed in [2], but in a randomised setting. We give schemes to efficiently transform a probabilistic algorithm for generating central elements into one for approximately counting ideals. In turn, we show how to use this approximate counting of ideals to bootstrap the original generating procedure into one with a strongly amplified probability of generating a central element. Finally, in section 5, we close by listing some avenues for further research.

2 Definitions and Preliminaries

Let $P := (P, \leq)$ be a finite poset. A subset I of P is called an *order-ideal* or simply, *ideal* of P if it is downwards-closed: $y \leq x \in I$ implies $y \in I$. We denote the total number of ideals of P by N_P , and the number of ideals containing a given element $x \in P$ by $N_P(x)$.

For $\delta > 0$, an element $x \in P$ is called δ -*central* in P if

$$\delta \leq N_P(x)/N_P \leq 1 - \delta.$$

3 Generalised Searching and Sorting in Posets

Let $P := (P, \leq)$ be a poset and $f : P \rightarrow R$ be an order-preserving map. The *searching* problem for P is: given $\alpha \in R$, determine if there is a $p \in P$ such that $f(p) = \alpha$ using as few comparisons of the form $f(x) = \alpha$ or $f(x) \leq \alpha$ for $x \in P$ as possible. This is clearly a generalisation of the usual search problem in linear orders.

A direct information-theoretic argument shows that $\log N$ comparisons are, in general, necessary [5]. Linial and Saks [5], showed that $O(\log N)$ comparisons are also sufficient. This is achieved via their remarkable

Theorem 1 (Central Element Theorem, [5]). *In every finite poset P , there exists a δ -central element c such that*

$$\delta \leq \frac{N(c)}{N(P)} \leq 1 - \delta$$

where $\delta \geq \delta_0 := \frac{1}{4}(3 - \log_2 5) \approx 0.17$.

Briefly, at any stage, searching for α in a sub-poset Q of P , we always compare α with a δ_0 -central element in the Q (which is guaranteed to exist by the above), and then, depending on the result of the comparison, pass to the appropriate sub-poset of Q . This search procedure clearly makes $O(\log N)$ comparisons in the worst case.

For a poset $P := (P, \leq)$ where P is a known finite set, but \leq is unknown, the *sorting* problem is to determine the relation \leq using as few comparisons of the form $x \leq y$ for $x, y \in P$. This is clearly a generalisation of the usual sorting problem in linear orders.

Theorem 2 (Generalised Sorting in Posets). ⁴ *Let $P := (P, \leq)$ be a partial order with P a known finite set, but \leq unknown. Then the number of comparisons needed to determine the relation \leq is $\Theta(|P| \cdot \log N)$ where N is the number of ideals in P .*

Proof. The upper bound follows easily from the corresponding bound for the search problem. Given $P := \{p_1, \dots, p_n\}$, we successively determine the posets $P_i := (P_i, \leq_i)$, where $P_i := \{p_1, \dots, p_i\}$ and \leq_i is \leq restricted to P_i , for $1 \leq i \leq n$. Knowing the poset P_i we determine the poset P_{i+1} by *inserting* p_{i+1} in the correct position relative to P_i using the search procedure from above. This requires $O(\log N(P_i)) = O(\log N)$ comparisons at each stage, giving a total of $O(|P| \cdot \log N)$ comparisons overall.

Observe that $n+1 \leq N \leq 2^n$. We denote by \mathcal{P}_N the set of partial orders with n elements and at most N ideals. An algorithm which solves the general sorting problem is a ternary tree. Each node (apart from the leaves) has tree branchings corresponding to the possible results ($\leq, >, \parallel$) of a comparison. Therefore we need at least $\log_3 |\mathcal{P}_N|$ questions to sort all posets in \mathcal{P}_N .

In the following we show that $\log_3 |\mathcal{P}_N| = \Omega(n \log N)$. This is obvious if $N \leq n^2$ since \mathcal{P}_N contains all $n!$ many linear orders. Otherwise ($N > n^2$) we consider a subclass \mathcal{S}_N of \mathcal{P}_N and show that it contains at least $2^{\Omega(n \log N)}$ posets.

Let $m = \frac{\log N}{4}$, $l = \lfloor \frac{n}{m} \rfloor$. We partition the ground set V into l subsets V_1, V_2, \dots, V_l of cardinality m and one subset V_0 of cardinality $n - ml$. The subclass \mathcal{S}_N consists of all partial orders which contain the relations $x < y$ if $x \in V_i, y \in V_j$ and $i + 2 \leq j$.

To show that $\mathcal{S}_N \subseteq \mathcal{P}_N$ we remark that each ideal I of a partial order in \mathcal{S}_N is determined by its maximal elements. If j is the greatest index such that I contains an element y of V_j then y is a maximal element of I and all other

⁴ Theorem 2 was also obtained independently by Hannah Baumgarten (personal communication).

maximal elements of I can be chosen from the set $V_j \cup V_{j-1}$. Hence each poset in \mathcal{S}_N contains at most $\frac{n}{m} 2^{2m} \leq N$ ideals.

Now we have to count the number of posets in \mathcal{S}_N . The relations between the elements of V_i and the elements of V_{i+1} are undetermined for each i and can be chosen independently. Therefore there are at least $(2^m)^{m(l-1)} (2^m)^{n-m^l} = 2^{m(n-m)}$ posets in \mathcal{S}_N . Since $m = \frac{\log N}{4} \leq \frac{n}{4}$ we get the lower bound $2^{\Omega(n \log N)}$.

Remark. 1. By a result of Shannon, a tree with arity a and l leaves has average path length at least $\log l / \log a$ [8]. This, together with a result of Yao, implies that the lower bound holds also for randomized algorithms [10].

2. Both the previous propositions are existence results since they require the use of δ -central elements, and the proof of [5] merely asserts the existence of such elements without showing how to find them explicitly. For special classes of posets, algorithms are known to produce such central elements explicitly and then this yields constructive results [9, 2].
3. Faigle and Turan present in [3] an explicit sorting algorithm for general posets that uses $O(\min\{w(P), \log n\} n \log N)$ comparisons where $w(P)$ denotes the width of the poset P . Their incremental algorithm determines in each round i a minimum chain cover of the suborder P_i , which is already known and inserts a new element (whose relations to the elements in P_i are undetermined) into each chain. We remark that insertion into a chain of length m requires at most $2 \log(m+1) + 1$ comparisons (one needs two binary searches in a chain C if the new element is not comparable with all elements of C). By Dilworth's theorem there are at most $w(P)$ such chains and hence the algorithm needs at most

$$C(P) = 2n w(P) \left(\frac{1}{2} + \log(n + w(P)) \right) - \log w(P)$$

comparisons. Since $w(P) \leq \log N$ and $n \leq N$ we can bound the number of comparisons by

$$C(P) = O(\min\{w(P), \log n\} n \log N).$$

4 Central Elements and Ideal Counting

For the searching and sorting problems on posets, it is clear that central elements play a key role – if one can find central elements efficiently, then the two problems can be solved with an optimal number of comparisons.

In [2], the authors presented evidence to the effect that the problem of finding central elements explicitly may be a difficult one for general posets. Specifically, they showed that given an efficient algorithm to generate a central element in a poset, one can obtain an efficient algorithm to *approximately* count the number of ideals in the poset. Provan and Ball, [6] show that the problem of *exactly* counting the number of ideals in a poset is #P-complete.

In this section, we pursue the relation between the problem of generating central elements and the ideal-counting problem further, specifically, in a randomised setting. The motivation is two-fold. First, for some classes of posets, there are easy or even trivial ways of generating a central element with high probability. Second, we would like to explore if randomisation helps in significantly reducing the complexity of the approximation version of a #P-complete problem, as indeed it demonstrably does in some well-documented cases, [4, 1].

We begin with a simple lemma which was also used in [2].

Lemma 3.

1. If $P := P_1 \times \dots \times P_n$, then $N(P) = N(P_1) \times \dots \times N(P_n)$.
2. For any poset P , let P_l denote the lifted poset consisting of an anti-chain of l elements below P (in the terminology of [2], this is the series composition of an anti-chain of l elements and P). Then,

$$N(P_l) = N(P) + 2^l - 1$$

$$N_{P_l}(x) = \begin{cases} N_P(x) & \text{if } x \in P \\ N(P) + 2^{l-1} - 1 & \text{if } x \in P_l \setminus P \end{cases}$$

Corollary 4. Let P, P_l be as in lemma 3 and $N = N(P)$. Then for any $\delta > 0$,

1. if $2^l < r_1 := \frac{2\delta}{1-2\delta}(N-1)$ then only elements from P can be δ -central for P_l ,
2. if $2^l > r_2 := \frac{1-\delta}{\delta}N + 1$ then only elements from $P_l \setminus P$ can be δ -central for P_l .

For l such that $r_1 < 2^l < r_2$ elements from both P and $P_l \setminus P$ can be central (Figure 1).

Proof. From lemma 3 it follows that if $x \in P_l \setminus P$ is δ -central for P_l then

$$\frac{N(P) + 2^{l-1} - 1}{N(P) + 2^l - 1} \leq 1 - \delta$$

$$\Rightarrow 2^l \geq \frac{2\delta}{1-2\delta}(N-1)$$

This establishes part 1. Similarly we can prove the second part.

Proposition 5. Given any $\delta > 0$ and a probabilistic procedure $A(P)$ that generates elements from P such that $P[A(P) \text{ is } \delta\text{-central}] \geq \frac{3}{4}$, there is a $t > 0$ that depends only on δ , and a probabilistic procedure $B(P)$ that outputs a number \tilde{N} such that $P[1/t \leq \tilde{N}/N(P) \leq t] \geq 1 - 1/4^{|P|}$. Moreover if A runs in polynomial time so does B .

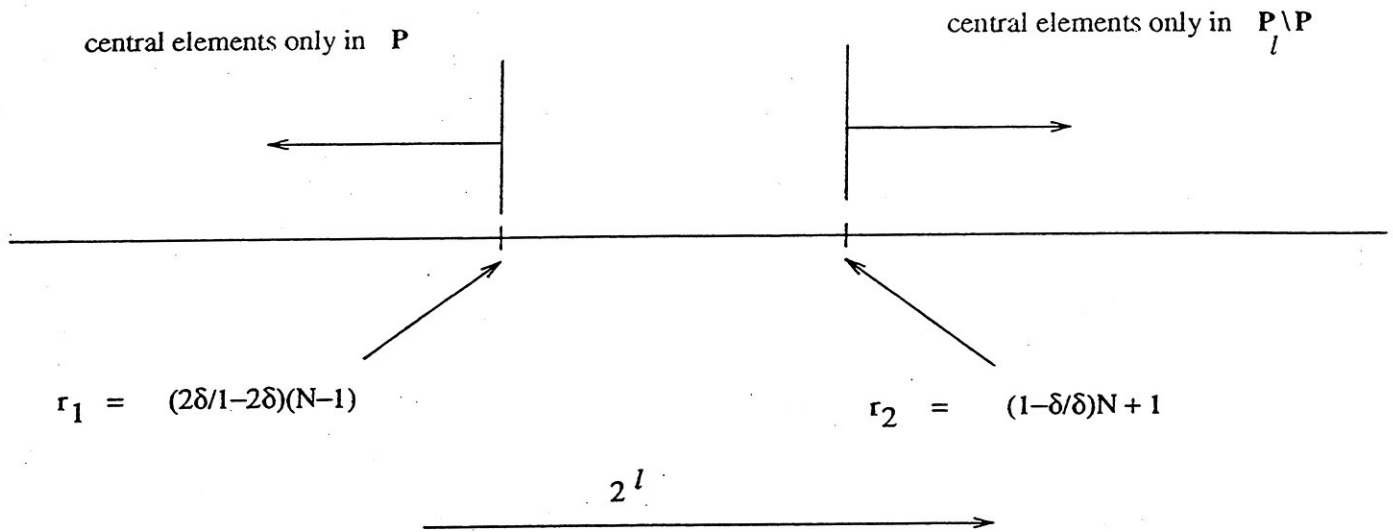


Fig. 1. Central Elements of P_l

Algorithm $B(P)$

```

l ← 0
repeat
  l ← l + 1
  if  $2^l > \frac{1-\delta}{\delta} * 2^{|P|}$  then return  $2^l$  and QUIT
  /* This is to make sure that  $\mathcal{A}$  always halts in polynomial time */
  Run  $\mathcal{A}(P_l)$   $2m + 1$  times. Let  $A_i$  denote the output of the  $i$ th run.
  if  $|\{i \mid A_i \in P\}| \leq m$  then
    return  $2^l$  and QUIT
forever

```

Fig. 2. Algorithm B for approximating the number of ideals

Proof. We claim that the algorithm described in figure 2 does the job. Invoking corollary 4 if $2^l < r_1$ only elements from P can be δ -central. Hence with probability at least $\frac{3}{4}$, \mathcal{A} returns an element from $P_l \setminus P$. Then using a standard computation (see lemma 3.4 in [7] for instance), one can show that the probability that procedure B outputs any specific $2^l < r_1$ is at most $\frac{1}{2}(\frac{3}{4})^m$. It follows that $P[B(P) > r_1] \geq 1 - (\lceil \log r_1 \rceil) \frac{1}{2}(\frac{3}{4})^m$. Similarly, one can show that $P[B(P) < 2r_2] \geq 1 - \frac{1}{2}(\frac{3}{4})^m$. Then choosing $t = \frac{2(1-\delta)}{\delta}$ and appropriate m (for example, $m = 4|P|^2 + 1/(1-2\delta)$ suffices) the proposition is satisfied. Note that

the running time of B is polynomial if that of A is.

Approximations to ideal counts can be dramatically improved by using the first construction in Lemma 3. Applying algorithm B to the poset P^s (for appropriately chosen s) as in [2], we observe that the output satisfies the inequalities $1/t \leq (\tilde{N}/N(P))^s \leq t$ with the stated high probability. Choosing s to make $t^{1/s} \leq 1 + \epsilon$ and $t^{-1/s} \geq 1 - \epsilon$, we obtain the claimed amplification:

Theorem 6. *Given a probabilistic procedure $A(P)$ that generates elements from P such that $P[A(P) \text{ is } \delta\text{-central}] \geq \frac{3}{4}$, and $\alpha > 0$, there is a probabilistic algorithm $C_\alpha(P)$ that outputs a number \tilde{N} such that $P[1 - \alpha \leq \tilde{N}/N(P) \leq 1 + \alpha] > 1 - 1/2^{|P|}$.*

Moreover if A runs in polynomial time so does C .

This accurate ideal-counting procedure can be employed to construct a procedure that with over-whelming probability accepts when an *arbitrary* element in poset is *sufficiently* central, and rejects when the element is not, starting with a procedure that *generates* a central element with high probability. More precisely,

Proposition 7. *Given any $\delta > 0$ and a probabilistic procedure $A(P)$ that generates elements from P such that $P[A(P) \text{ is } \delta\text{-central}] \geq \frac{3}{4}$, for any $\delta_0, \epsilon > 0$ one can construct a procedure $\mathcal{D}_\epsilon(P, x)$ such that*

- x is (δ_0) -central $\Rightarrow P[\mathcal{D}_\epsilon(P, x) = 1] \geq 1 - 1/2^{|P|}$,
- x is not $(\delta_0 - \epsilon)$ -central $\Rightarrow P[\mathcal{D}_\epsilon(P, x) = 1] < 1/2^{|P|}$.

Proof. Our procedure \mathcal{D} applies procedure $C_\alpha(P)$ (with $\alpha = \epsilon/20$ which is obtained from the procedure A as above) to estimate the ideals containing the given element x , $N(x)$ and to estimate the total number of ideals, $N(P)$. It returns 1 iff the ratio of these two numbers is between $(\delta_0 - \beta)$ and $1 - (\delta_0 - \beta)$ that is if it deems element x to be $(\delta_0 - \beta)$ -central where $\beta = \epsilon/5$.

Note: Actually, the procedure does not work correctly with high probability only for elements x such that $(\delta_0 - \epsilon) \leq N(x)/N(P) \leq (\delta_0 - \beta)$ or $1 - (\delta_0 - \beta) \leq N(x)/N(P) \leq 1 - (\delta_0 - \epsilon)$.

Interestingly, this test procedure enables one to bootstrap the original generation procedure to construct one with a strongly amplified probability of generating a central element.

Theorem 8. *Given any $\delta, \epsilon > 0$ and a procedure $A(P)$ that outputs elements of P with $P[A(P) \text{ is } \delta\text{-central}] > 3/4$, one can construct a procedure $\mathcal{E}_\epsilon(P)$ such that $P[\mathcal{E}_\epsilon(P) \text{ is } (\delta_0 - \epsilon)\text{-central}] > 1 - 1/2^{|P|}$, for any δ_0 such that there are δ_0 -central elements in P . Moreover, if A runs in polynomial time so does \mathcal{E} .*

Proof. Procedure \mathcal{E} applies the test \mathcal{D} to all elements of P and outputs the first element for which \mathcal{D} returns 1.

One can show that with very high probability, this procedure returns an element that is $(\delta_0 - \epsilon)$ -central.

5 Conclusions and Open Problems

For the class of posets of constant height, an element chosen uniformly at random is central with some constant probability. It remains to be seen if this probability can be amplified to be bounded away from $\frac{1}{2}$ and if this can be extended to simple probabilistic algorithms for other classes of posets.

Alternatively, do there exist probabilistic approximation schemes to count the number of ideals in posets? In particular can the approach via rapidly-mixing Markov chains from [4, 1] be applied to this problem?

An interesting property of the central element problem is that for some classes of posets it seems easier to probabilistically generate central elements than to test if a given element is central. It is an open question when the existence of a probabilistic generation procedure implies the existence of a probabilistic testing procedure. In particular, is this the case for the central element problem?

6 Acknowledgements

Shiva Chaudhuri pointed out that Yao's result implied the randomised lower-bound in section 3. We thank the anonymous referees for their comments and suggestions for improvements to the paper.

References

1. M. Dyer, A. Frieze and R. Kannan: *A Random Polynomial Time Algorithm for Approximating the Volume of Convex Bodies* J. ACM 38:1 1–17 (1991).
2. U. Faigle, L. Lovász, R. Schrader and Gy. Turán: *Searching in Trees, Series-Parallel and Interval Orders*, SIAM J. Computing, 12:4, 1075–1084, (1986).
3. U. Faigle and Gy. Turán: *Sorting and Recognition Problems for Ordered Sets*, SIAM J. Computing, 17:1, 100–113, (1988).
4. M. Jerrum and A. Sinclair: *Approximating the Permanent* SIAM J. Computing 18:6, 1149–1178 (1989).
5. N. Linial and M. Saks: *Every Poset has a Central Element*, J. Comb. Theory, Series A 40, 195–210 (1985).
6. J.S. Provan and M.O. Ball: *The Complexity of Counting Cuts and Computing the Probability that a Graph is Connected* SIAM J. Computing 12 777–788 (1983).
7. U. Schöning: *Complexity and Structure* Lecture Notes in Computer Science, 211, Springer-Verlag Berlin Heidelberg New York, (1985)
8. C.E. Shannon: *A mathematical theory of communication*, Bell System Tech. Journal 27, 379–423 and 623–656 (1948)
9. G. Steiner: *Searching in 2-Dimensional Partial Orders*, Journal of Algorithms 8, 95–105, (1987).
10. A.C.C. Yao *Probabilistic Computations: Towards a Unified Measure of Complexity* 18th Annual Symp. on Foundations of Comp. Sci., 222–227 (1977)