SIAM J. COMPUT.
Vol. 17, No. 1, February 1988

SORTING AND RECOGNITION PROBLEMS FOR ORDERED SETS*

U. FAIGLE† AND GY. TURÁN‡

Abstract. How many questions are needed to decide whether an unknown ordered set is isomorphic to a fixed ordered set P_0? This recognition problem is considered, together with some related computational problems concerning ordered sets.

Key words. sorting, recognition, identification, order, complexity, algorithm

AMS(MOS) subject classifications. 68E05, 68C25

Introduction. The standard sorting problem is to identify an unknown order knowing that it is a linear order. This formulation suggests the following generalization to ordered sets, called the *P_0-sorting problem*: determine an unknown order, knowing that it is isomorphic to a fixed "pattern" order P_0 (thus P_0 is a chain in the classical case).

A related question is the *P_0-recognition problem*: decide whether the unknown order is isomorphic to P_0. The *identification problem* asks for determining the unknown order without any a priori information.

In this paper we discuss the complexity of these problems. The model of computation is the usual decision tree model with the only modification that now every node has three sons since two elements can turn out to be incomparable as well. For the sorting and recognition problems we use the worst-case measure of complexity. This measure is of no use for the identification problem as clearly every identification algorithm has to ask every pair in order to identify an antichain. Instead, we introduce a refined measure of complexity which assigns a function $C_A(P)$ to every algorithm A giving the worst-case behavior of A on orders isomorphic to P for every order P.

In § 2 we observe some connections between the complexities of the above problems. We need the notion of an *essential set* (consisting of covering and critical pairs), essentially defined in Rabinovitch and Rival [8] and Kelly [4], to describe these connections.

An example is given of an adversary argument to bound the sorting complexity in § 3. In § 4 the recognition complexity of Boolean algebras is determined to be $\Theta(n \log_2 n)$. It is shown that the minimal recognition complexity of orders with height 1 on n elements is asymptotically $n \log_2 n$. (Also, an $\Omega(n \log_2 n)$ lower bound holds for ordered sets with width less than $n^{1-\varepsilon}$ for any $\varepsilon > 0$, in particular for orders with bounded width.)

Section 5 contains identification algorithms. The first uses Dilworth decomposition, the second is based on a recent result of Linial and Saks [6] on the existence of central elements; both generalize sorting by insertion. These algorithms are optimal for orders of bounded width but they can perform badly in general. The third algorithm is optimal for orders of height 1, the "other extreme." The merging of these algorithms is still not optimal as is shown by their behavior on Boolean algebras.

* Received by the editors February 5, 1985; accepted for publication (in revised form) April 20, 1987. This research was supported by the joint research project "Algorithmic Aspects of Combinatorial Optimization" of the Hungarian Academy of Sciences (Magyar Tudományos Akadémia) and the German Research Association (Deutsche Forschungsgemeinschaft, SFB 21).

† Institut fur Ökonometrie und Operations Research, Abteilung Operations Research, Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn, West Germany. Present address, Department of Applied Mathematics, University of Twente, Enschede, Netherlands.

‡ Department of Mathematics, Statistics and Computer Science, University of Illinois, Chicago, Illinois 60637. This author was on leave from Automata Theory Research Group of the Hungarian Academy of Sciences, Szeged, Hungary.

Some remarks and open problems are given in § 6. In particular, it is pointed out that the problems discussed are special cases of more general problems which are ordered set versions of graph property recognition problems in the sense discussed in Bollobás [1].

1. Definitions. Ordered sets P on n elements are assumed to have a fixed ground set $\{a_1, \cdots, a_n\}$; the order relation is $<$; incomparability is denoted by $\|$. An element x is *isolated* if $x\|y$ for every y. The pair (a_i, a_j) is a *covering pair* if $a_i < a_j$ and there is no a_k with $a_i < a_k < a_j$ (these pairs form the Hasse diagram of P). The *width* $w(P)$ (resp. the *height* $h(P)$) of P is the size of a maximum antichain in P (resp. the length of a maximum chain in P).

An *ideal* I (resp. a *filter* F) is a subset of elements subject to $x \in I$, $y < x$, which implies $y \in I$ (resp. $x \in F$, $y > x$ implies $y \in F$). $I_P(x) = \{y: y < x\}$ is the ideal generated by x, $F_P(x) = \{y: y > x\}$ denotes the filter generated by x. N_P is the total number of ideals in P and $N_P(x)$ is the number of ideals containing x.

The theorem of Dilworth [2] relates the width of P to a chain cover of the ground set: a chain cover of P with k chains exists if and only if $k \geqq w(P)$.

Let P_0 be an ordered set on n elements.

The *P_0-sorting problem* is to determine an unknown order P knowing only that P is isomorphic to P_0.

An *algorithm* A to solve the P_0-sorting problem is a *ternary decision tree* with nonleaves labeled "$a_i : a_j$" for some $1 \leqq i < j \leqq n$ and outgoing edges labeled "$a_i < a_j$," "$a_i \| a_j$," "$a_i > a_j$." A leaf is either labeled by an order $P \simeq P_0$ or by a sign xx. If it is labeled by P, then P is the only order isomorphic to P_0 consistent with the answers obtained along the path leading to that leaf. If it is labeled by xx, then there is no order isomorphic to P_0 satisfying the answer on the corresponding path. For $P \simeq P_0$, $A(P)$ is the number of questions used to find P (the length of the unique path leading to the leaf labeled P).

The *complexity* of A is

$$C_A^s := \max \{A(P): P \simeq P_0\}$$

and

$$C^s(P_0) := \min \{C_A^s: A \text{ is a } P_0\text{-sorting algorithm}\}$$

is the *sorting complexity* of P_0.

The *P_0-recognition problem* is to decide if an unknown order P is isomorphic to P_0.

An *algorithm* A to solve the recognition problem is again a ternary tree with nonleaves and edges labeled as above. Leaves are labeled with "yes" or "no." If a leaf is labeled "yes," then all orders satisfying the answers given along the path leading to the leaf must be isomorphic to P_0. If it is labeled "no," then there can be no order isomorphic to P_0 consistent with these answers.

The *complexity* C_A^r of A is the depth of the tree and

$$C^r(P_0) := \min \{C_A^r: A \text{ is a } P_0\text{-recognition algorithm}\}$$

is the *recognition complexity* of P_0.

The *identification problem* is to determine an unknown order without any a priori information.

An *algorithm* A to solve the identification problem is a sequence $A = (A_1, A_2, \cdots)$, where A_n is an algorithm to solve the identification problem on n elements. Thus A_n is a ternary tree as above, with leaves labeled by orders on $\{a_1, \cdots, a_n\}$ subject to a leaf labeled P if P is the only ordered set consistent with the answers on the corresponding path.

$A(P)$ is the number of questions used by A_n to identify P and

$$C_A(P) := \max \{A(P'): P' \simeq P\}$$

is the *complexity* of A to identify orders isomorphic to P. Note that here the complexity of A is a function.

An identification algorithm is defined to be *optimal* if for every identification algorithm B

$$C_A(P) = O(C_B(P)).$$

(Notation: $f(P) = O(g(P))$ means $|f(P)/g(P)| \leqq \alpha$ for some constant $\alpha > 0$, $f(P) = \Omega(g(P))$ if $g(P) = O(f(P))$ and $f(P) = \Theta(g(P))$ if $f(P) = O(g(P))$ and $f(P) = \Omega(g(P))$, $f(n) = o(g(n))$ if $f(n)/g(n) \to 0$ as $n \to \infty$.)

If P is an ordered set on $\{a_1, \cdots, a_n\}$, then $E = E_1 \cup E_2 \subseteq \{a_1, \cdots, a_n\} \times \{a_1, \cdots, a_n\}$ is an *essential set* for P if

(a) $(a_i, a_j) \in E_1$ implies $a_i < a_j$ and $(a_i, a_j) \in E_2$ implies $a_i \| a_j$,

(b) P is the only order satisfying (a).

(Thus E is a set of comparabilities and incomparabilities which uniquely determines P.) The size of a minimum essential set is denoted by $e(P)$.

2. Some observations.

In this section we collect some useful facts about the problems defined above.

LEMMA 1. *For every order P_0,*

$$\max (C^s(P_0), e(P_0)) \leqq C^r(P_0) \leqq C^s(P_0) + e(P_0).$$

Proof. Consider a P_0-recognition algorithm and apply it to an order $P \simeq P_0$. Then arriving at the leaf, P must be identified. Indeed, if the relation between a_i and a_j were not determined then it is easy to see that $a_i \| a_j$ and, say, $a_i > a_j$ would both be consistent. It is impossible, however, that both relations yield orders isomorphic to P_0 (i.e., the number of incomparable pairs is not the same). Thus every recognition algorithm can be used for sorting and hence $C^s(P_0) \leqq C^r(P_0)$. Furthermore, $e(P_0) \leqq C^r(P_0)$ because in the above case enough questions must be asked to determine P uniquely.

A recognition algorithm for P_0 can be obtained as follows: assume that the unknown order is isomorphic to P_0 and apply a sorting algorithm. Then check the elements of a minimal size essential set to justify the assumption. Thus $C^r(P_0) \leqq C^s(P_0) + e(P_0)$. □

LEMMA 2. *If* A *is an identification algorithm then for every order P_0,*

$$C_A(P_0) \geqq C^r(P_0).$$

Proof. Every identification algorithm can be used as a P_0-recognition algorithm: run A for $C_A(P_0)$ steps. Then every $P \simeq P_0$ is already identified. If the order is not identified yet it cannot be isomorphic to P_0. □

LEMMA 3. *An identification algorithm* A *is optimal iff*

$$C_A(P_0) = O(C^r(P_0)).$$

Proof. ($\leftarrow$) Follows from Lemma 2 above.

($\rightarrow$) If $C_A(P_0)/C^r(P_0)$ is not bounded, choose, for every k, an order P_k on n_k elements with $C_A(P_k) > k \cdot C^r(P_k)$ such that $n_1 < n_2 < \cdots$. Let B be an identification algorithm obtained from A by merging A_{n_k} with an optimal recognition algorithm for

P_k (merging means asking questions of the two algorithms alternatingly). It follows from the proof of Lemma 1 that B in fact is an identification algorithm and $C_B(P_k) \leqq 2C^r(P_k)$. Thus $C_A(P_k) > (k/2)C_B(P_k)$ and A is not optimal. (Also note that B is a "better" algorithm as $C_B(P) \leqq 2C_A(P)$ holds for every P.) □

The next lemma is a reformulation of observations made by Rabinovitch and Rival [8] and Kelly [4].

A pair (a_i, a_j) is *critical* if $a_i \| a_j$, $x > a_i$ implies $x > a_j$, and $y < a_j$ implies $y < a_i$.

LEMMA 4. *For every P_0, the covering and critical pairs form the unique minimal essential set.*

Proof. These pairs are necessary as their status is not implied by the other relations. Covering pairs imply all comparabilities. Incomparability of critical pairs implies all incomparabilities. (See [8], [4].) □

In the sequel this unique minimal essential set will be denoted by $E(P_0)$.

3. Sorting. There are simple examples with small sorting complexity. For example, if P_0 has a unique maximal element and all other elements are incomparable, then $C^s(P_0) = \lceil (n-1)/2 \rceil$.

If $\#(P_0)$ denotes the number of orders isomorphic to P_0 on the ground set $\{a_1, \cdots, a_n\}$ then clearly

$$C^s(P_0) \geqq \log_3 \#(P_0).$$

Partitioning the elements into antichains according to their height (i.e., the length of the longest chain ending with them) and considering the sizes $s_1, \cdots, s_n$ of these antichains we get

$$\#(P_0) \geqq \frac{n!}{s_1! \cdots s_n!}$$

and with $s_i \leqq w(P_0)$ this implies

$$C^s(P_0) \geqq n \log_3 n - n \log_3 w(P_0) - 5n.$$

Later we shall see (Corollary 18 in § 5) that

$$C^s(P_0) \leqq w(P_0) \cdot 2n \log_2 n + 3nw(P_0).$$

Thus, when w is fixed, the sorting complexity is determined within a constant factor.

Another obvious example where the information-theoretic bound is not sharp, besides the example above, is given by the n-element order containing only one comparable pair. Here $\#(P_0) = n(n-1)$ and $C^s(P_0) = \binom{n}{2}$.

A special case which may be of some interest is the order consisting of independent comparable pairs (a "matching"), where the information-theoretic bound is $\Omega(n \log_2 n)$.

The following theorem determines the sorting complexity of a matching. The upper bound was observed by M. Aigner.

THEOREM 5. *Let n be even and P_0 be the parallel composition of $n/2$ 2-element chains. Then for the sorting complexity of P_0*

$$C^s(P_0) = \frac{n^2}{4}.$$

Proof. For the lower bound we use the "greedy" adversary strategy: assume that at every stage of the algorithm the answer to a question "$a_i : a_j$" is always "$a_i \| a_j$" unless adding this incomparability to the information provided already leaves no order isomorphic to P_0 consistent with these data. In the latter case the answer is "$a_i > a_j$." (It is easy to see that the direction of the inequality in the answer is arbitrary; it is only fixed for being definite.)

Assume that the algorithm stops by determining the order $P \simeq P_0$ with comparable pairs $a_1 > a_2, a_3 > a_4, \cdots, a_{n-1} > a_n$. Then it is clear that each of these pairs must have been asked by the algorithm as otherwise reversing one inequality would also be consistent with every other answer. Letting $E(>)$ denote the set of these comparable pairs, we assume that they have been asked by the algorithm in the above order. Further, let $E(\|)$ be the set of pairs (a_i, a_j) asked by the algorithm with answer "$a_i \| a_j$" obtained and $\bar{E}$ be the set of unasked pairs.

Just before (a_1, a_2) was asked, the set of pairs not asked yet contained $E(>) \cup \bar{E}$ and as the adversary uses the greedy strategy, it must have been the case that every perfect matching (corresponding to the comparable pairs) in this set of pairs contains (a_1, a_2). Hence every perfect matching in $E(>) \cup \bar{E}$ contains (a_1, a_2).

The same argument shows that every perfect matching in $E(>) \cup \bar{E}$ contains $(a_3, a_4), \cdots, (a_{n-1}, a_n)$ as well, thus $E(>) \cup \bar{E}$ contains exactly one perfect matching.

Now we use the following result of Hetyei [3] (see also [7, Problem 7.24]):

LEMMA 6 [3]. *If a graph on n vertices has exactly one perfect matching then it has at most $n^2/4$ edges.*

This implies $|E(\|)| \geqq n(n-2)/4$. As all pairs in $E(\|)$ and $E(>)$ have been asked by the algorithm, the lower bound follows.

A sorting algorithm can be given as follows: starting with an arbitrary element a_1, the only element a_2 comparable to it can be found with $n-1$ questions. Elements a_1 and a_2 can then be discarded. To find the next comparable pair, $n-3$ questions are needed, etc. Altogether $(n-1)+(n-3)+\cdots+1 = n^2/4$ questions are required.

4. Recognition. In this section we first determine the recognition complexity of Boolean algebras (this example will be used in the next section). Then we show that the minimal recognition complexity of orders with height 1 on n elements is asymptotically $n \log_2 n$. Finally we construct orders of height 4 with linear size essential set (implying that the proof for height 1 does not extend to arbitrary bounded height).

In the next section it is shown (Corollary 18) that

$$C^r(P_0) \leqq w(P_0) \cdot 2n \log_2 n + 3nw(P_0)$$

and, if $h(P_0) = 1$, then

$$C^r(P_0) \leqq 2e(P_0).$$

This means that the recognition complexity is determined for orders of bounded width or of height 1 up to a constant factor. (Note that we do not have an analogous statement for the sorting complexity of orders of height 1.)

THEOREM 7. *Let $n = 2^m$ and B_n be the Boolean algebra on m atoms. Then for the recognition complexity of B_n*

$$C^r(B_n) = \Theta(n \log_2 n).$$

LEMMA 8. $e(B_n) = (n/2+1) \log_2 n$ (if $m > 2$).

Proof. We use Lemma 4. Every element has degree m in the Hasse diagram. Critical pairs form a matching of atoms and co-atoms. □

LEMMA 9. *If P is an arbitrary ordered set on n elements, then a maximal (nonextendible) chain can be found in $n \cdot \lceil \log_2 (h(P)+2) \rceil$ steps.*

Proof. Assume we found a chain $x_1 < \cdots < x_k$. Take an unused element y and start inserting y into the chain by binary search. If the search is successful, y is inserted in at most $\lceil \log_2 (k+1) \rceil$ steps. If $y \| x_i$ for some $1 \leqq i \leqq k$ then the chain cannot be extended by y and hence y can be excluded from later comparisons. □

LEMMA 10. $C^s(B_n) \leqq n(\log_2 n + 2\lceil \log(\log n + 2)\rceil + 2)$.

Proof. We describe a B_n-sorting algorithm.

Find a maximal chain $x_0 < \cdots < x_m$ in $n\lceil \log(\log n + 2)\rceil$ steps. Then x_1 is an atom. Compare every element to x_1: those incomparable to x_1 form a Boolean algebra with $m-1$ atoms. Repeat the same process altogether m times. Finally the atoms $y_1, \cdots, y_m$ are identified and we used at most $2n(1 + \lceil \log(\log n + 2)\rceil)$ steps.

Now compare every element with $y_1, \cdots, y_m$. This determines the Boolean algebra since

$$u \geqq v \quad \text{iff } v \geqq y_i \text{ implies } u \geqq y_i \quad \text{for every atom } y_i.$$

The last phase needs at most $n \log_2 n$ steps. □

Proof of Theorem 7. The upper bound follows directly from Lemmas 1, 8 and 10. By fixing atoms $b_1, \cdots, b_m$ all the other elements are already determined. Thus

$$C^r(B_n) \geqq C^s(B_n) \geqq \log_3 \#(B_n) \geqq \log_3 \left(\binom{n}{m} (n-m)! \right) = n \log_3 n + o(n \log n)$$

$$= \Omega(n \log_2 n). \qquad □$$

We now turn to orders of height 1.

LEMMA 11. *For every $\varepsilon > 0$ there exists an n_0 such that if P_0 is an ordered set on n elements with $n \geqq n_0$ and $h(P_0) = 1$ then*

$$C^r(P_0) \geqq (1-\varepsilon) n \log_2 n.$$

For the proof of Lemma 11, we use the following lemma.

LEMMA 12. *For every $\varepsilon > 0$ there exists an n_0 such that if P_0 is an ordered set on n elements with $n \geqq n_0$ and $h(P_0) = 1$ then*

$$e(P_0) \geqq (1-\varepsilon) n \log_2 n.$$

Proof. Let $P_0 = A \cup B$ where the elements in A are maximal and the elements in B are minimal (isolated elements are distributed arbitrarily). Assume $|A| = l$, $|B| = k$, $k \leqq \lfloor n/2 \rfloor$. Then every pair (x, y) with $x \in B$, $y \in A$ belongs to $E(P_0)$ (every such pair is either critical or covering). Consider now $x, y \in A$. Then $x \| y$ and (x, y) or (y, x) is a critical pair iff $I_{P_0}(x)$ and $I_{P_0}(y)$ are comparable. If $I_{P_0}(x) = I_{P_0}(y)$ then (x, y) is critical.

Let $H_1, \cdots, H_t$ $(t = 2^k)$ be the subsets of B and s_i $(I \leqq i \leqq t)$ be the number of elements x in A with $I_{P_0}(x) = H_i$. Then there are

$$r = \sum_{i=1}^{t} \binom{s_i}{2}$$

pairs (x, y) with $I_{P_0}(x) = I_{P_0}(y)$ $(x, y \in A, x \neq y)$. As $\sum_{i=1}^{t} s_i = l$, the convexity of the function $f(x) = x^2$ implies that the above sum is minimized if all the s_i are

$$\lfloor l/2^k \rfloor \quad \text{or} \quad \lceil l/2^k \rceil.$$

Thus

$$r \geqq \binom{\lfloor l/2^k \rfloor}{2} \cdot 2^k.$$

Hence

$$\begin{aligned} e(P_0) &\geqq k\cdot l+\binom{\lfloor l/2^k\rfloor}{2}\cdot 2^k \\ &\geqq k\cdot l+\binom{\lfloor n/2^{k+1}\rfloor}{2}\cdot 2^k \\ &\geqq k\cdot l+\frac{(n-2^{k+1})(n-2^{k+2})}{8\cdot 2^k}. \end{aligned}$$

If

$$n \geqq 4\cdot 2^{k+1} \tag{$*$}$$

then

$$e(P_0)\geqq k\cdot l+\frac{1}{32}\cdot\frac{n^2}{2^k}.$$

We distinguish two cases. If

$$k\geqq \log_2 n-\log_2\log_2 n-5$$

then

$$e(P_0)\geqq(\log_2 n-\log_2\log_2 n-5)(n-\log_2 n)\geqq(1-\varepsilon)n\log_2 n$$

when n is sufficiently large. If

$$k<\log_2 n-\log_2\log_2 n-5$$

then $(*)$ holds and

$$2^k<\frac{n}{32\log_2 n}.$$

Thus

$$e(P_0)\geqq\frac{1}{32}\cdot\frac{n^2}{2^k}>n\log_2 n. \qquad \square$$

Proof of Lemma 11. Using Lemma 1, the lemma follows immediately from Lemma 12 above. □

LEMMA 13. *For every $\varepsilon>0$, there exists an n_0, such that if $n\geqq n_0$ then there exists an order P_0 on n elements with $h(P_0)=1$ and recognition complexity*

$$C_r(P_0)\leqq(1+\varepsilon)n\log_2 n.$$

The proof is given in the next section after Corollary 22.

THEOREM 14. *Let $f(n):=\min\{C^r(P_0)\colon |P_0|=n,\ h(P_0)=1\}$. Then*

$$\frac{f(n)}{n\log_2 n}\to 1 \quad \text{as } n\to\infty.$$

Proof. Follows directly from Lemmas 11 and 13. □

Lemma 12 might suggest that $e(P_0)=\Omega(n\log_2 n)$ for orders with $h(P)\leqq k$ for every fixed k. The next theorem shows that this is not true.

THEOREM 15. *There exist orders of height* 4 *with $e(P)=(4+o(1))|P|$.*

Proof. Take two disjoint sets A, B with $|A|=|B|=2m$ and define

$\mathscr{H}_1 := \{\{x\}: x \in A \cup B\}$;

$\mathscr{H}_2 := \{H: |H| = m \text{ and } H \subseteq A \text{ or } H \subseteq B\}$;

$\mathscr{H}_3 := \{H: H = H_1 \cup H_2 \text{ with } |H_1| = |H_2| = m, H_1 \subseteq A, H_2 \subseteq B\}$;

$\mathscr{H}_4 := \{H: |H| = 3m, (A \cup B) - H \in \mathscr{H}_2\}$;

$\mathscr{H}_5 := \{A \cup B - \{x\}: x \in A \cup B\}$;

$\mathscr{H} := \bigcup_{i=1}^5 \mathscr{H}_i$;

$P_{\mathscr{H}} := \mathscr{H}$ ordered by inclusion.

Then clearly $P_{\mathscr{H}}$ is of height 4 and has

$$8m + 4\binom{2m}{m} + \binom{2m}{m}^2 = \binom{2m}{m}^2 (1 + o(1))$$

elements. Furthermore, the Hasse-diagram of $P_{\mathscr{H}}$ contains

$$8m\binom{2m-1}{m-1} + 4\binom{2m}{m}^2 = 4\binom{2m}{m}^2 (1 + o(1))$$

edges. The first term counts edges incident to maximal and minimal elements. The second term counts edges incident to $\mathscr{H}_3$: each $H \in \mathscr{H}_3$ contains exactly two sets in $\mathscr{H}_2$ and is contained in exactly two sets from $\mathscr{H}_4$.

The critical pairs in $P_{\mathscr{H}}$ are the pairs $(A \cup B - \{x\}, \{x\})$ as in the case of the full Boolean algebra.

Thus

$$e(P) = 4\binom{2m}{m}^2 (1 + o(1)),$$

proving the theorem. $\square$

(Note that if we delete some elements of $\mathscr{H}_3$ so that each set in $\mathscr{H}_2$ and $\mathscr{H}_4$ has still neighbors in $\mathscr{H}_3$, the necessary properties of $P_{\mathscr{H}}$ are preserved. Thus for every n there exist height 4 orders with linear size essential set.)

To close this section we note that from the bound given at the beginning of § 3 it follows that

$$C^r(P_0) = \Omega(n \log_2 n)$$

if $w(P_0) \leqq n^{1-\varepsilon}$ for any $\varepsilon > 0$, suggesting problem 2 in the last section.

5. Identification. We describe some identification algorithms and give bounds for their performance.

The first two algorithms generalize sorting by insertion. Inserting an element x into an ordered set P' means to determine which elements are smaller, larger or incomparable to x.

Both algorithms A and B proceed in n stages on orders on $\{a_1, \cdots, a_n\}$. After stage i the suborder P_i on $\{a_1, \cdots, a_i\}$ of P is determined. Stage $i+1$ consists of inserting a_{i+1} into P_i. We only give the insertion methods.

Insertion in algorithm A. Choose a minimum chain cover $C_1, \cdots, C_{l_i}$ of P_i and insert a_{i+1} into each chain. Insertion into a chain is done by a straightforward modification of binary search.

To describe algorithm B we note that the elements of $\{a_1, \cdots, a_i\}$ that are smaller than a_{i+1} in P form an ideal I_i in P_i (resp. the elements which are larger form a filter F_i of P_i).

Insertion in algorithm B. Determine I_i and F_i by algorithm C below.

Obviously, the two tasks are equivalent so we only formulate the algorithm to determine I_i.

An element z of P' is called a *central element* if for every z' in P'

$$\left|\frac{N_{P'}(z)}{N_{P'}}-\frac{1}{2}\right| \leqq \left|\frac{N_{P'}(z')}{N_{P'}}-\frac{1}{2}\right|.$$

Algorithm C below determines the ideal $P'(x)=\{y: y \in P', y<x\}$ for an already identified order P' and a new element x recursively.

ALGORITHM C.
Given P' and x choose a central element z of P' and compare it with x.
If $x>z$ then put $P'(x) := I_{P'}(z) \cup \{z\} \cup P''(x)$, where $P''=P'-(I_{P'}(z)\cup\{z\})$.
If $x \not> z$ then put $P'(x) := P''(x)$, where $P''=P'-(F_{P'}(z)\cup\{z\})$.

We need the following result of Linial and Saks [6].

LEMMA 16 [6]. *If z is a central element then*

$$\delta_0 \leqq \frac{N_{P'}(z)}{N_{P'}} \leqq 1-\delta_0,$$

where

$$\delta_0 = \tfrac{1}{4}(3-\log_2 5) \simeq 0.17.$$

THEOREM 17. A *and* B *are correct identification algorithms. For every ordered set P it holds that*

$$C_A(P) \leqq 2nw(P)(\tfrac{1}{2}+\log_2(n+w(P))-\log_2 w(P))$$

and

$$C_B(P) \leqq 7.46 n \log_2 N_P.$$

Proof. The correctness of A is obvious. The correctness of B follows from observing that if $x>z$ then $x>z'$ for every $z'<z$ in P' and if $x \not> z$ then $x \not> z'$ for every $z'>z$ in P'.

The bound for A holds since insertion into a chain of length m is easily seen to require at most $2\log_2(m+1)+1$ steps and at each stage $l_i \leqq w(P)$ by Dilworth's theorem. (The computation uses the concavity of $\log_2(x)$.)

The bound for B is obtained as in [6] noting that the choice of a central element always reduces the number of ideals by a factor of $(1-\delta_0)$ at least, using Lemma 16 above. □

COROLLARY 18. *For every order P_0*

$$C^s(P_0) \leqq C^r(P_0) \leqq w(P_0)2n \log_2 n + 3nw(P_0).$$

Proof. Immediate from Lemmas 1 and 2. □

Comparing A and B we note that A is simpler to compute. The bound of B is not worse than that of A but we do not know of any examples where it performs much better.

Now we describe another identification algorithm, efficient for orders of height 1. We use the following variables:

P'	denotes the actual order obtained up to a certain stage of the algorithm (including the comparabilities and incomparabilities deduced);
MAX	(resp. MIN) is the set of elements that are maximal (resp. minimal) in P'; (where now x is maximal (resp. minimal) if $x \not< z$ for every z and $x > y$ for some y (resp. $x \not> z$ and $x < y$ for some y));
ISO	is the set of elements that are known to be isolated in P (thus $x \in$ ISO iff it is already known to be incomparable to every element);
U	is the set of "unprocessed" elements ($U = P - (\text{MAX} \cup \text{MIN} \cup \text{ISO})$).

A pair (x, y) is called *undetermined* in P' if neither $x \| y$, $x > y$ nor $x < y$ is in P'; otherwise it is determined.

ALGORITHM D.

```
1.  $P' := \varnothing$, MAX $:= \varnothing$, MIN $:= \varnothing$, ISO $:= \varnothing$, $U := \{a_1, \cdots, a_n\}$.
2.  If $U = \varnothing$ compare every undetermined pair $x, y \in$ MAX subject to $I_P, (x) \subseteq$
    $I_P, (y)$ and every undetermined pair $x, y \in$ MIN subject to $F_P, (x) \subseteq F_p, (y)$,
    go to 6.
3.  Choose a $z \in U$ and compare it to every $x$ subject to $(z, x)$ is undetermined
    in $P'$. Update $P'$, MAX, MIN, ISO, $U$. If $h(P') > 1$, go to 5.
4.  If $|\text{MAX}| \leqq |\text{MIN}|$ compare every $x \in$ MAX with every $y$ s.t. $(x, y)$ is undeter-
    mined in $P'$. Update $P'$, MAX, MIN, ISO, $U$. If $h(P') > 1$ go to 5.
    If $|\text{MIN}| \leqq |\text{MAX}|$ compare every $x \in$ MIN with every $y$ s.t. $(x, y)$ is un-
    determined in $P'$. Update $P'$, MAX, MIN, ISO, $U$. If $h(P') > 1$ go to 5,
    else go to 2.
5.  Compare every undetermined pair.
6.  Stop.
```

Informally, the algorithm proceeds as follows. Sets MAX, MIN and ISO contain elements already known to be maximal, minimal or isolated. If there is an element z not contained in any of them, it is compared to every element. To update necessary information, elements in the smaller of the two new sets MIN and MAX (or elements in both sets, if they are of equal size) are compared to every other element. If MAX ∪ MIN ∪ ISO contains every element, all remaining undetermined pairs are compared. If at any point the order turns out to have height > 1, all pairs are compared.

PROPOSITION 19. *Algorithm* D *is a correct identification algorithm.*

Proof. If $h(P') > 1$ during an execution of steps 3 or 4 then every undetermined pair is compared in step 5. So the order is obviously identified. Otherwise the algorithm reaches the execution of step 2: after an execution of step 3 z belongs to MAX, MIN or ISO. So $|U|$ decreases.

If step 2 is reached then the elements in ISO are already compared with every other element. Furthermore, every pair (x, y) with $x \in$ MAX, $y \in$ MIN is determined as can be seen from the previous execution of step 4. Thus the only undetermined pairs (x, y) must satisfy either $x, y \in$ MAX or $x, y \in$ MIN (we have $P = \text{MAX} \cup \text{MIN} \cup$

ISO at that point). If $x, y \in \mathrm{MAX}$ (resp. $x, y \in \mathrm{MIN}$) and $I_{P'}(x)$ and $I_{P'}(y)$ are incomparable sets then (x, y) is determined as it must be true that $x \| y$. Therefore all undetermined pairs are compared in step 2 and the order is identified indeed. □

THEOREM 20. *If* $h(P)=1$ *then* $C_D(P) \leqq 2e(P)$.

Proof. Let P be an order of height 1. Since every pair is compared at most once, it is sufficient to show that the number of nonessential pairs ever compared by the algorithm is at most $e(P)$.

The definition of critical pairs implies that every pair compared in step 2 is essential. (In fact, only pairs from the larger of the two sets MAX, MIN are compared at that step.) If in step 3 the element z turns out to be isolated, then every pair containing z is critical and in this case MAX and MIN are not modified.

Therefore, the algorithm can compare nonessential pairs (x, y) only during those executions of steps 3 and 4 where the element z turns out to be maximal or minimal. (Furthermore, for every such pair either both x, y are maximal or both are minimal.) An execution of steps 3 and 4, where z turns out to be maximal or minimal, is called a *phase*.

Let MAX_i, MIN_i, ISO_i and U_i denote the values of the sets MAX, MIN, ISO and U after phase i. Let z_i be the element selected in phase i and V_i be the set of elements comparable to z_i. □

LEMMA 21. *For every* i:

(a) *if* $|\mathrm{MAX}_i| \leqq |\mathrm{MIN}_i|$, *then after phase i every pair containing an element from* MAX_i *is determined*;

(b) *if* $|\mathrm{MIN}_i| \leqq |\mathrm{MAX}_i|$, *then after phase i every pair containing an element from* MIN_i *is determined.*

Furthermore, there are sets MAX_i^*, MIN_i^* *with the following properties*:

(c) $\mathrm{MAX}_i^* \cup \mathrm{MIN}_i^*$ *covers all pairs* (u, v) *compared by the algorithm in phases* $1, \cdots, i$ *subject to* $\{u, v\} \cap \mathrm{ISO}_i = \varnothing$;

(d) $|\mathrm{MAX}_i^* \cup \mathrm{MIN}_i^*| \leqq 2 \min(|\mathrm{MAX}_i|, |\mathrm{MIN}_i|)$;

(e) $\mathrm{MAX}_i^* \subseteq \mathrm{MAX}_i$, $\mathrm{MIN}_i^* \subseteq \mathrm{MIN}_i$;

(f) *if* $|\mathrm{MAX}_i| \leqq |\mathrm{MIN}_i|$ *then* $\mathrm{MAX}_i^* = \mathrm{MAX}_i$;

(g) *if* $|\mathrm{MIN}_i| \leqq |\mathrm{MAX}_i|$ *then* $\mathrm{MIN}_i^* = \mathrm{MIN}_i$.

Proof. By induction on i. For $i=0$ all sets are empty. We distinguish several cases.

Case 1. $|\mathrm{MAX}_i| < |\mathrm{MIN}_i|$.

Case 1.1. z_{i+1} is a maximal element.

Then in step 3 of phase $i+1$, z_{i+1} is added to MAX and V_{i+1} is added to MIN. (It may be the case that $V_{i+1} \subseteq \mathrm{MIN}_i$.)

Case 1.1.1. $|\mathrm{MAX}_i \cup \{z_{i+1}\}| < |\mathrm{MIN}_i \cup V_{i+1}|$.

Then using (a) by induction, no further comparisons are performed in step 4 of phase $i+1$. Putting $\mathrm{MAX}_{i+1}^* := \mathrm{MAX}_i^* \cup \{z_{i+1}\}$, $\mathrm{MIN}_{i+1}^* := \mathrm{MIN}_i^*$, all conditions remain satisfied.

Case 1.1.2. $|\mathrm{MAX}_i \cup \{z_{i+1}\}| = |\mathrm{MIN}_i \cup V_{i+1}|$.

(This may occur only if $V_{i+1} \subseteq \mathrm{MIN}_i$.) In this case, no further comparisons are performed in the first "if" of step 4 of phase $i+1$, but there may be comparisons involving elements from MIN_i in the second "if" of step 4 of phase $i+1$. However, $\mathrm{MIN}_{i+1} = \mathrm{MIN}_i$, $\mathrm{MAX}_{i+1} \subseteq \mathrm{MAX}_i \cup \{z_{i+1}\}$ and thus $|\mathrm{MIN}_{i+1}| \leqq |\mathrm{MAX}_{i+1}|$, so $\mathrm{MAX}_{i+1}^* := \mathrm{MAX}_i^* \cup \{z_{i+1}\}$, $\mathrm{MIN}_{i+1}^* := \mathrm{MIN}_i$ satisfies the requirements.

Case 1.2. z_{i+1} is a minimal element.

By (a), $V_{i+1} \cap \mathrm{MAX}_i = \varnothing$, otherwise z_{i+1} would be contained in MIN_i. So in step 3 of phase $i+1$, new maximal element(s) will be added to MAX.

Case 1.2.1. $|\mathrm{MAX}_i \cup V_{i+1}| < |\mathrm{MIN}_i \cup \{z_{i+1}\}|$.

Then in step 4 of phase $i+1$, all undetermined pairs containing an element from V_{i+1} are compared and no other pairs. We can choose $\mathrm{MAX}^*_{i+1} := \mathrm{MAX}^*_i \cup V_{i+1}$, $\mathrm{MIN}^*_{i+1} := \mathrm{MIN}^*_i \cup \{z_{i+1}\}$.

Case 1.2.2. $|\mathrm{MAX}_i \cup V_{i+1}| = |\mathrm{MIN}_i \cup \{z_{i+1}\}|$.

In the first "if" of step 4 of phase $i+1$, all undetermined pairs involving an element from V_{i+1} are compared. Possibly all undetermined pairs involving an element from MIN_i are compared in the second "if" of step 4. We can choose $\mathrm{MAX}^*_{i+1} := \mathrm{MAX}^*_i \cup V_{i+1}$, $\mathrm{MIN}^*_{i+1} := \mathrm{MIN}_i \cup \{z_{i+1}\}$.

Case 1.2.3. $|\mathrm{MAX}_i \cup V_{i+1}| > |\mathrm{MIN}_i \cup \{z_{i+1}\}|$.

Then in step 4 of phase $i+1$, undetermined pairs containing an element from MIN_i are compared and we can choose $\mathrm{MAX}^*_{i+1} := \mathrm{MAX}^*_i$, $\mathrm{MIN}^*_{i+1} := \mathrm{MIN}_i \cup \{z_{i+1}\}$.

Case 2. $|\mathrm{MAX}_i| = |\mathrm{MIN}_i|$.

Case 2.1. z_{i+1} is a maximal element.

By (b) of the induction hypothesis, $V_{i+1} \cap \mathrm{MIN} = \varnothing$ and we get two cases.

Case 2.1.1. $|V_{i+1}| = 1$.

Let $V_{i+1} = \{z'_{i+1}\}$, then all undetermined pairs involving z'_{i+1} are compared in the second "if" of step 4 of phase $i+1$. We can choose $\mathrm{MAX}^*_{i+1} := \mathrm{MAX}^*_i \cup \{z_{i+1}\}$, $\mathrm{MIN}^*_{i+1} := \mathrm{MIN}^*_i \cup \{z'_{i+1}\}$.

Case 2.1.2. $|V_{i+1}| > 1$.

In this case no further comparisons are performed in step 4 of phase $i+1$ and we can put $\mathrm{MAX}^*_{i+1} := \mathrm{MAX}^*_i \cup \{z_{i+1}\}$, $\mathrm{MIN}^*_{i+1} := \mathrm{MIN}^*_i$.

The remaining cases require analogous arguments, so we give the definitions of the sets MAX^*_{i+1}, MIN^*_{i+1} only.

Case 2.2. z_{i+1} is a minimal element.

Case 2.2.1. $|V_{i+1}| = 1$.

$$\mathrm{MAX}^*_{i+1} := \mathrm{MAX}^*_i \cup \{z'_{i+1}\}, \qquad \mathrm{MIN}^*_{i+1} := \mathrm{MIN}^*_i \cup \{z_{i+1}\}.$$

Case 2.2.2. $|V_{i+1}| > 1$.

$$\mathrm{MAX}^*_{i+1} := \mathrm{MAX}^*_i, \qquad \mathrm{MIN}^*_{i+1} := \mathrm{MIN}^*_i \cup \{z_{i+1}\}.$$

Case 3. $|\mathrm{MAX}_i| > |\mathrm{MIN}_i|$.

(Note that step 4 is not symmetric in MAX and MIN.)

Case 3.1. z_{i+1} is a maximal element.

Case 3.1.1. $|\mathrm{MAX}_i \cup \{z_{i+1}\}| > |\mathrm{MIN}_i \cup V_{i+1}|$.

$$\mathrm{MAX}^*_{i+1} := \mathrm{MAX}^*_i \cup \{z_{i+1}\}, \qquad \mathrm{MIN}^*_{i+1} := \mathrm{MIN}^*_i \cup V_{i+1}.$$

Case 3.1.2. $|\mathrm{MAX}_i \cup \{z_{i+1}\}| = |\mathrm{MIN}_i \cup V_{i+1}|$.

$$\mathrm{MAX}^*_{i+1} := \mathrm{MAX}_i \cup \{z_{i+1}\}, \qquad \mathrm{MIN}^*_{i+1} := \mathrm{MIN}^*_i \cup V_{i+1}.$$

Case 3.1.3. $|\mathrm{MAX}_i \cup \{z_{i+1}\}| < |\mathrm{MIN}_i \cup V_{i+1}|$.

$$\mathrm{MAX}^*_{i+1} := \mathrm{MAX}_i \cup \{z_{i+1}\}, \qquad \mathrm{MIN}^*_{i+1} := \mathrm{MIN}^*_i.$$

Case 3.2. z_{i+1} is a minimal element.

Case 3.2.1. $|\mathrm{MAX}_i \cup V_{i+1}| > |\mathrm{MIN}_i \cup \{z_{i+1}\}|$.

$$\mathrm{MAX}^*_{i+1} := \mathrm{MAX}^*_i, \qquad \mathrm{MIN}^*_{i+1} := \mathrm{MIN}^*_i \cup \{z_{i+1}\}.$$

Case 3.2.2. $|\mathrm{MAX}_i \cup V_{i+1}| = |\mathrm{MIN}_i \cup \{z_{i+1}\}|$.

$$\mathrm{MAX}^*_{i+1} := \mathrm{MAX}_i, \qquad \mathrm{MIN}^*_{i+1} := \mathrm{MIN}^*_i \cup \{z_{i+1}\}.$$

Returning to the proof of Theorem 20, let i be the last phase, $\text{MAX}^* := \text{MAX}_i^*$ and $\text{MIN}^* := \text{MIN}_i^*$. The rest of the argument is symmetric in MAX and MIN, so assume $|\text{MAX}| \leqq |\text{MIN}|$. Then Lemma 21 implies $\text{MAX}^* = \text{MAX}$ and as $|\text{MAX}^*| + |\text{MIN}^*| \leqq 2|\text{MAX}|$, also $|\text{MIN}^*| \leqq |\text{MAX}^*|$ holds. All nonessential pairs compared by the algorithm are either pairs of maximal elements or pairs of minimal elements covered by MIN*. So their number is at most

$$\binom{\text{MAX}}{2} + \left(\binom{\text{MIN}}{2} - \binom{\text{MIN} - \text{MIN}^*}{2}\right)$$

$$\leqq \binom{\text{MAX}}{2} + \binom{\text{MIN}}{2} - \binom{\text{MIN} - \text{MAX}}{2} = \text{MAX} \cdot (\text{MIN} - 1)$$

$$< \text{MAX} \cdot \text{MIN} \leqq e(P). \qquad \square$$

(We remark that the example of a matching shows that the factor 2 in the bound is sharp for the algorithm.)

COROLLARY 22. *If $h(P) = 1$ then $C_D(P) \leqq 2 \cdot C^r(P)$ and $C^r(P) \leqq 2 \cdot e(P)$.*

Proof. Immediate from Lemmas 1 and 2. $\square$

Now we prove Lemma 13 of previous section.

Proof of Lemma 13. We construct the order P_0 as follows: P_0 has $\lceil (1+\varepsilon) \log_2 n \rceil$ maximal elements and $n - \lceil (1+\varepsilon) \log_2 n \rceil$ minimal elements. Each minimal element has $\lceil (1+\varepsilon)/2 \cdot \log_2 n \rceil$ upper covers so that the set of upper covers is different for each minimal element. This condition is easy to satisfy as there are

$$\binom{\lceil (1+\varepsilon) \log_2 n \rceil}{\left\lceil \frac{1+\varepsilon}{2} \log_2 n \right\rceil} \geqq c_\varepsilon \cdot \frac{n^{1+\varepsilon}}{\sqrt{\log_2 n}}$$

choices. Furthermore, it can also be required that the set of lower covers be different for each maximal element and each be of size between, e.g., $n/4$ and $3n/4$. Then

$$e(P_0) = \lceil (1+\varepsilon) \log_2 n \rceil (n - \lceil (1+\varepsilon) \log_2 n \rceil).$$

Thus a $2(1+\varepsilon) n \log_2 n$ recognition algorithm for P_0 is given by running D for $C_D(P_0)$ steps. (If the order is not identified up to this point, it cannot be isomorphic to P_0.)

A slight modification gives the smaller upper bound. A pair $x > y$ can be found in $n-1$ steps; y can be compared to everything in at most $(n-2)$ steps. Then y must have $\lceil (1+\varepsilon)/2 \log_2 n \rceil$ upper covers (otherwise $P \not\simeq P_0$). Comparing the upper covers of y to everything we get every minimal element except at most one. Comparing the remaining at most $\lceil (1+\varepsilon)/2 \log_2 n \rceil + 1$ elements to everything, P is identified whenever $P \simeq P_0$. $\square$

Finally we point out that algorithms B and D are not sufficient to produce an optimal algorithm. Let algorithm E be obtained by merging B and D (in the sense of the proof of Lemma 3).

Then for the Boolean algebras B_n we have

$$C^r(B_n) = O(n \log_2 n)$$

from Theorem 7 and

$$C_E(B_n) = \Omega\left(\frac{n^2}{\log_2 n}\right).$$

Indeed, if the antichain of size

$$\binom{m}{m/2} = \Omega\left(\frac{m}{\sqrt{\log_2 n}}\right)$$

is examined first then every pair is compared in order to perform the insertion in B, and D is of no help in this case.

6. Some open problems. (1) It would be interesting to find good bounds for the sorting complexity in general.

(2) The special cases discussed in § 4 suggest that perhaps the recognition complexity is $\Omega(n \log n)$ for every order P_0.

(3) Is it true that there is no optimal identification algorithm in the strong sense of optimality defined here (cf. Lemma 3)? Are there other measures of complexity for identification algorithms?

(4) The definition of sorting and recognition problems introduced here is actually a special case of a more general one. One can consider sorting and recognition problems for arbitrary classes of orders (in this paper we considered classes that consist of isomorphic copies of a fixed order P_0). It seems reasonable to assume that these classes are closed under isomorphism, i.e., they are *ordered set properties.* In this framework the identification problem is just the sorting problem with respect to the class of all orders. The recognition problems arising this way are natural ordered set versions of analogous problems concerning graphs (see Bollobás [1, Chap. 8] for an excellent survey). The ordered set properties are substantially different as there are many "easy" properties with complexity $o(n^2)$ like having a unique maximal element, being a linear order, having bounded width or setup number. Nevertheless it is not difficult to give $\Omega(n^2)$ lower bounds, e.g., for connectivity, bounded height, being a lattice, an interval order, etc. What properties can be proved to be "elusive," i.e., to have complexity $\binom{n}{2}$?

Acknowledgment. We are grateful to M. Aigner for his remarks (in particular for pointing out the upper bound in Theorem 5) and to a referee for suggesting a simplification of the proof of Theorem 20.

Note added in proof. There are some further results concerning problems 2 and 3. It was shown that interval orders have recognition complexity $\Omega(n \log n)$ and there is an optimal identification algorithm for semiorders (U. Faigle and Gy. Turán, *The complexity of interval orders and semiorders,* Discrete Math., to appear). It was also shown that almost all orders have recognition complexity $\Omega(n \log n)$ (H. J. Prömel, *Counting unlabeled structures,* J. Combin. Theory Ser. A, 44 (1987), pp. 83-93). Recently M. Saks solved problem 2. In general, he showed that every n element order has recognition complexity at least $\frac{2}{5}n \log_3 n + o(n \log n)$ (M. Saks, *Recognition problems for transitive relations,* to be published).

REFERENCES

[1] B. Bollobás (1978), *Extremal Graph Theory,* Academic Press, New York.
[2] R. P. Dilworth (1950), *A decomposition theorem for partially ordered sets,* Annals Math., 51, pp. 161-166.
[3] G. Hetyei (1964), unpublished.
[4] D. Kelly (1981), *On the dimension of partially ordered sets,* Discrete Math., 35, pp. 135-156.
[5] D. E. Knuth (1973), *The Art of Computer Programming, Vol. 3, Sorting and Searching,* Addison-Wesley, Reading, MA.
[6] N. Linial and M. E. Saks (1983), *Information bounds are good for search problems on ordered structures,* Proc. 24th Annual IEEE Conference on the Foundations of Computer Science, pp. 473-475.
[7] L. Lovász (1979), *Combinatorial Problems and Exercises,* Akadémiai Kiadó.
[8] L. Rabinovitch and L. Rival (1979), *The rank of a distributive lattice,* Discrete Math., 25, pp. 275-279.