Recursive Sorting in Lattices

Jens Gerlach
Fraunhofer FOKUS, Berlin, Germany

Abstract

The direct application of the definition of sorting in lattices [1] is impractical because it leads to an algorithm with exponential complexity. In this paper we present for distributive lattices a recursive formulation to compute the sort of a sequence. This alternative formulation is inspired by the identity \( \binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k} \) that underlies Pascal’s triangle. It provides quadratic complexity and is in fact a generalization of insertion sort for lattices.

1 Background

If someone asked whether there is for the numbers \( x \) and \( y \) and the exponent \( n \) a general relationship between the value \( (x+y)^n \) and the powers \( x^n \) and \( y^n \), then the (obvious) answer is that this relationship is captured by the Binomial Theorem

\[
(x + y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k
\]

which also shows that other powers of \( x \) and \( y \) are involved.

If, on the other hand, \( (x_1, \ldots, x_n) \) is a sequence in a totally ordered set \( (X, \leq) \) and someone asked whether there is a general relationship between the elements of \( x \) and the elements of its nondecreasingly sorted counterpart \( x^\uparrow = (x_1^\uparrow, \ldots, x_n^\uparrow) \), then one could provide an easy answer for the first and last elements of \( x^\uparrow \). In fact, we know that \( x_1^\uparrow \) is the least element of \( \{x_1, \ldots, x_n\} \)

\[
x_1^\uparrow = x_1 \land \ldots \land x_n = \bigwedge_{k=1}^{n} x_k,
\]

whereas \( x_n^\uparrow \) is the greatest element of \( \{x_1, \ldots, x_n\} \)

\[
x_n^\uparrow = x_1 \lor \ldots \lor x_n = \bigvee_{k=1}^{n} x_k.
\]
The relationship between an arbitrary element \( x^\uparrow_k \) and the elements of \( x \) reads

\[
x^\uparrow_k = \bigwedge_{I \in \mathbb{N}\binom{n}{k}} \bigvee_{i \in I} x_i
\]

and has been proven in a previous work of the author [1, Proposition 2.2]. Here \( \mathbb{N}\binom{n}{k} \) is the set of subsets of \([1, n]\) that contain exactly \( k \) elements. Note that \( \mathbb{N}\binom{n}{k} \) consists of \( \binom{n}{k} \) elements.

Equation (1) is not just a compact formula for computing \( x^\uparrow \). It also provides a way to generalize the concept of sorting beyond totally ordered sets. In fact, if \((X, \leq)\) is a partially ordered set, that is also a lattice \((X, \wedge, \vee)\), then for each finite subset \( A \) of \( X \) both the infimum and supremum of \( A \) exist (denoted by \( \wedge A \) and \( \vee A \), respectively). Thus, the right hand side of Equation (1) is well-formed in a lattice. Therefore we can define (as in [1, Definition 3.1]) for \( x = (x_1, \ldots, x_n) \) a new sequence \( x^\triangle = (x^\triangle_1, \ldots, x^\triangle_n) \) by

\[
x^\triangle_k := \bigwedge_{I \in \mathbb{N}\binom{n}{k}} \bigvee_{i \in I} x_i.
\]

We refer to \( x^\triangle \) as \( x \) sorted with respect to the lattice \((X, \wedge, \vee)\). It can be shown that this definition of sorting in lattices maintains many properties that are familiar from sorting in totally ordered sets. For example, the sequence \((x^\triangle_1, \ldots, x^\triangle_n)\) is nondecreasing [1, Lemma 2.3] and the mapping \( x \mapsto x^\uparrow \) is idempotent [1, Lemma 3.6].

Note that we reserve the notation \( x^\uparrow \) in order to refer to sorting in totally ordered sets whereas we use the notation \( x^\triangle \) to refer to sorting in lattices.

2 The need for a more efficient formula

Definition (2) is nice and succinct, but it is also are quite impractical to use in computations. While conducting some experiments with Equation (2) in the lattice \((\mathbb{N}, \gcd, \lcm)\) it became obvious that only for very short sequences \( x \) the sequence \( x^\triangle \) can be computed in a reasonable time.

Table 1 shows simple performance measurements (conducted on a notebook computer) for computing \((1, \ldots, n)^\triangle\) in \((\mathbb{N}, \gcd, \lcm)\). The reason for this dramatic slowdown is of course the exponential complexity inherent in Equation (2):

In order to compute \( x^\triangle \) from \( x \) it is necessary to consider all \( 2^n - 1 \) nonempty subsets of \([1, n]\).
In order to address this problem, we prove in Proposition 1 the recursive Identity (8) for the case of bounded distributive lattices. This identity is closely related to the well-known fact that the binomial coefficient
\[
\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}
\]
can be efficiently computed through the recursion underlying Pascal’s triangle
\[
\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}.
\]
Furthermore, we prove in Proposition 2 that a lattice, in which Identity (8) holds, is necessarily distributive.

### 3 Recursive sorting in lattices

For the remainder of this paper we assume that \((X, \land, \lor, \bot, \top)\) is a bounded lattice. Here \(\bot\) is the least element of \(X\) and the neutral element of join
\[
x = \bot \lor x = x \lor \bot \quad \forall x \in X.
\]
At the same time, \(\top\) is the greatest element of \(X\) and the neutral element of meet
\[
x = \top \land x = x \land \top \quad \forall x \in X.
\]
We are now introducing a notation that allows us to concisely refer to individual elements of both \((x_1, \ldots, x_n)\) and \((x_1, \ldots, x_{n-1})\). Here again, it is convenient to employ the symbol for the binomial coefficient \(\binom{n}{k}\) in the context of sorting in lattices.

For a sequence \(x\) of length \(n\) we define for \(0 \leq m \leq n\)
\[
x^\land \left( \binom{m}{k} \right) := \begin{cases} 
\bot & k = 0 \\
(x_1, \ldots, x_m)^\land(k) & k \in [1, m] \\
\top & k = m + 1
\end{cases}
\]

| sequence length | 20  | 21  | 22  | 23  | 24  | 25  | 26  |
|-----------------|-----|-----|-----|-----|-----|-----|-----|
| time in s       | 0.6 | 1.3 | 2.7 | 5.8 | 11.8| 25.5| 51.6|

Table 1: Wall-clock time for computing \((1, \ldots, n)\) according to Equation (2)
We know from the definition of $x^\Delta$ in Equation (2) that
\[
(x_1, \ldots, x_m)^\Delta(k) = \bigwedge_{i \in I(k)} \bigvee x_i
\]
holds for $1 \leq k \leq m$. We therefore have for $1 \leq k \leq m$
\[
x^\Delta\left(\frac{m}{k}\right) = \bigwedge_{i \in I(k)} \bigvee x_i.
\] (6)

In particular, the identity
\[
x^\Delta\left(\frac{n}{k}\right) = x_k^\Delta
\] (7)
holds for $1 \leq k \leq n$.

The main result of this paper is Proposition 1, which states in Identity (8), how the $k$th element of $(x_1, \ldots, x_n)^\Delta$ can be computed from $(x_1, \ldots, x_{n-1})^\Delta$ and $x_n$ by simply applying one join and one meet.

The proof of Proposition 1 relies on the fact that the lattice under consideration is both bounded and distributive. The boundedness of $X$ is, in contrast to its distributivity, no real restriction because every lattice can be turned into a bounded lattice by adjoining a smallest and a greatest element [2, p. 7].

**Proposition 1.** If $(X, \land, \lor, \bot, \top)$ is a bounded distributive lattice and if $x$ is a sequence of length $n$, then
\[
x^\Delta\left(\frac{n}{k}\right) = x^\Delta\left(\frac{n-1}{k}\right) \land \left(x^\Delta\left(\frac{n-1}{k-1}\right) \lor x_n\right)
\] (8)
holds for $1 \leq k \leq n$.

**Proof.** We first consider the “corner cases” $k = 1$ and $k = n$.

For $k = 1$, we have
\[
x^\Delta\left(\frac{n}{1}\right) = \bigwedge_{i=1}^n x_i
\]
by Identity (6)
\[=
\bigwedge_{i=1}^{n-1} x_i \land x_n
\]
by associativity
\[= x^\Delta\left(\frac{n-1}{1}\right) \land x_n
\]
by Identity (6)
\[= x^\Delta\left(\frac{n-1}{1}\right) \land (\bot \lor x_n)
\]
by Identity (3)
\[= x^\Delta\left(\frac{n-1}{1}\right) \land (x^\Delta\left(\frac{n-1}{0}\right) \lor x_n)
\]
by Identity (5).
For $k = n$, we have
\[
\Delta^n (n) = \bigvee_{i=1}^n x_i \quad \text{by Identity (6)}
\]
\[
= \left( \bigvee_{i=1}^{n-1} x_i \right) \lor x_n \quad \text{by associativity}
\]
\[
= \Delta^{n-1} (n-1) \lor x_n \quad \text{by Identity (6)}
\]
\[
= \left( \Delta^{n-1} (n-1) \lor x_n \right) \land \top \quad \text{by Identity (4)}
\]
\[
= \Delta^{n-1} (n-1) \lor \left( \Delta^{n-1} (n-1) \lor x_n \right) \quad \text{by Identity (5)}
\]
\[
= \Delta^n (n) \land \left( \Delta^{n-1} (n-1) \lor x_n \right) \quad \text{by commutativity}.
\]

In the general case of $1 < k < n$, we first remark that if $A$ is a subset of $[1, n]$ that consists of $k$ elements, then there are two cases possible:

1. If $n$ does not belong to $A$, then $A$ is a subset of $\mathbb{N}^{(n-1)}_k$.

2. If $n$ is an element of $A$, then the set $B := A \setminus \{n\}$ belongs to $\mathbb{N}^{(n-1)}_{k-1}$.

In other words, $\mathbb{N}^{(n)}_k$ can be represented as the following (disjoint) union
\[
\mathbb{N}^{(n)}_k = \mathbb{N}^{(n-1)}_k \cup \left\{ B \cup \{n\} \mid B \in \mathbb{N}^{(n-1)}_{k-1} \right\}.
\]  
(9)

We conclude
\[
\Delta^n (n) = \bigwedge_{I \in \mathbb{N}^{(n)}_k} \bigvee_{i \in I} x_i \quad \text{by Identity (6)}
\]
\[
= \bigwedge_{I \in \mathbb{N}^{(n-1)}_k} \bigvee_{i \in I} x_i \land \bigwedge_{I \in \mathbb{N}^{(n-1)}_{k-1} \cup \{n\}} \bigvee_{i \in I \cup \{n\}} x_i \quad \text{by Identity (9)}
\]
\[
= \Delta^{n-1} (k) \land \bigwedge_{I \in \mathbb{N}^{(n-1)}_{k-1} \cup \{n\}} \bigvee_{i \in I \cup \{n\}} x_i \quad \text{by Identity (6)}
\]
which concludes the proof. □

The following Proposition 2 states that the converse of Proposition 1 also holds.

**Proposition 2.** Let \((X, \wedge, \lor, \bot, \top)\) be a bounded lattice which is not distributive. Then there exists a sequence \(x = (x_1, x_2, x_3)\) in \(X\) such that Identity (8) is not satisfied.

**Proof.** According to a standard result on distributive lattices [2, Theorem 4.7], a lattice is not distributive, if and only if it contains a sublattice which is isomorphic to either \(N_5\) or \(M_3\) (see Figure 1).

\[
\begin{align*}
\text{Figure 1: The non-distributive lattices } N_5 \text{ and } M_3
\end{align*}
\]

From Identity (2) (see also [1, Identity 9]) follows for the elements of \(x^\triangle = (x_1^\triangle, x_2^\triangle, x_3^\triangle)\)

\[
\begin{align*}
x_1^\triangle &= x_1 \land x_2 \land x_3 & (10a) \\
x_2^\triangle &= (x_1 \lor x_2) \land (x_1 \lor x_3) \land (x_2 \lor x_3) & (10b) \\
x_3^\triangle &= x_1 \lor x_2 \lor x_3. & (10c)
\end{align*}
\]
If $X$ contains the sublattice $N_5$, then we consider the sequence $x = (c, d, b)$ and its subsequence $(c, d)$. From Identity (10) then follows

$$(c, d, b)^\triangle = (a, d, e)$$

$$(c, d)^\triangle = (a, e).$$

Thus, we have

$$x^\triangle \left( \frac{3}{2} \right) = d \quad x^\triangle \left( \frac{3}{1} \right) = e \quad x^\triangle \left( \frac{3}{1} \right) = a.$$

However, applying Identity (8) we obtain

$$x^\triangle \left( \frac{3}{2} \right) = x^\triangle \left( \frac{3}{2} \right) \land \left( x^\triangle \left( \frac{3}{1} \right) \lor x_3 \right)$$

$$= e \land (a \lor b)$$

$$= e \land b$$

$$= b \quad \text{instead of } d.$$

If $X$ contains the sublattice $M_3$, then we consider the sequence $x = (b, c, d)$ and its subsequence $(b, c)$. From Identity (10) then follows

$$(b, c, d)^\triangle = (a, e, e)$$

$$(b, c)^\triangle = (a, e).$$

We therefore have

$$x^\triangle \left( \frac{3}{2} \right) = e \quad x^\triangle \left( \frac{3}{2} \right) = e \quad x^\triangle \left( \frac{3}{1} \right) = a.$$

Again, applying Identity (8) we obtain

$$x^\triangle \left( \frac{3}{2} \right) = x^\triangle \left( \frac{3}{2} \right) \land \left( x^\triangle \left( \frac{3}{1} \right) \lor x_3 \right)$$

$$= e \land (a \lor d)$$

$$= e \land d$$

$$= d \quad \text{instead of } e.$$

\[ \square \]

Using Identity (8), we can prove following Lemma 3 which generalizes a known fact known from sorting in a total order: If one knows that $x_n$ is greater or equal that the preceding elements $x_1, \ldots, x_{n-1}$ then sorting the sequence $(x_1, \ldots, x_n)$ can be accomplished by sorting $(x_1, \ldots, x_{n-1})$ and simply appending $x_n$.

**Lemma 3.** Let $(X, \land, \lor, \bot, \top)$ be a bounded distributive lattice and $x$ be a sequence of length $n$. If the condition $x_i \leq x_n$ holds for $1 \leq i \leq n - 1$, then

$$x^\triangle \left( \frac{n}{n} \right) = x_n$$

7
and
\[ x^\Delta \left( \frac{n}{i} \right) = x^\Delta \left( \frac{n-1}{i} \right) \]
holds for \( 1 \leq i \leq n - 1 \).

**Proof.** The first equation follows directly from the fact that \( x^\Delta_n \) is the supremum of the values \( x_1, \ldots, x_n \).

Regarding the second equation, we known from [1, Lemma 3.3] that if for \( 1 \leq i \leq n - 1 \) the inequality \( x_i \leq x_n \) holds, then
\[ x^\Delta \left( \frac{n-1}{i} \right) \leq x_n. \]
This inequality is also valid for \( i = 0 \) because
\[ x^\Delta \left( \frac{n-1}{0} \right) = \bot \]
holds by Identity (5). From general properties of meet and join then follows that
\[
\begin{align*}
  x^\Delta \left( \frac{n-1}{i} \right) \lor x_n &= x_n \\
  x^\Delta \left( \frac{n-1}{i} \right) \land x_n &= x^\Delta \left( \frac{n-1}{i} \right)
\end{align*}
\]
holds for \( 0 \leq i \leq n - 1 \). We can therefore simplify Identity (8)
\[ x^\Delta \left( \frac{n}{i} \right) = x^\Delta \left( \frac{n-1}{i} \right) \land \left( x^\Delta \left( \frac{n-1}{i-1} \right) \lor x_n \right) \]
first to
\[ = x^\Delta \left( \frac{n-1}{i} \right) \land x_n \]
and finally to
\[ = x^\Delta \left( \frac{n-1}{i} \right). \]
\[ \square \]

## 4 Insertion sort in lattices

Equation (11) symbolically represents Identity (8). Whenever an arrow \( \searrow \) and and arrow \( \nearrow \) meet, the values are combined by a meet. In the case of an arrow \( \searrow \), however, first the value at the origin of the arrow is combined with the sequence value \( x_n \) through a join.

\[ x_n \quad \searrow \quad x^\Delta \left( \frac{n}{k} \right) \]

\[ x^\Delta \left( \frac{n-1}{k-1} \right) \]

(11)
Figure 2 integrates several instances of Equation (11) in order to graphically represent Identity (8) and to emphasize its close relationship to Pascal’s triangle.

Figure 2: A graphical representation of Identity (8)

Figure 3 outlines an algorithm, which is based on Identity (8), and that starting from \((x_1)^\Delta = (x_1)\) successively computes \((x_1, \ldots, x_i)^\Delta\) for \(2 \leq i \leq n\).

\[
\begin{align*}
(x_1)^\Delta, x_2 &\mapsto (x_1, x_2)^\Delta \\
(x_1, x_2)^\Delta, x_3 &\mapsto (x_1, x_2, x_3)^\Delta \\
&\vdots \\
(x_1, \ldots, x_{n-1})^\Delta, x_n &\mapsto (x_1, \ldots, x_{n-1}, x_n)^\Delta
\end{align*}
\]

Figure 3: A simple algorithm based on Identity (8)

From Identity (8) follows that in step \(i\) exactly \(i\) joins and \(i\) meets must be performed. Thus, altogether there are

\[
\sum_{i=2}^{n} 2 \cdot i = n(n + 1) - 2
\]

applications of join and meet. In other words, such an implementation has quadratic complexity. The algorithm in Figure (3) can be considered as insertion sort [3, § 5.2.1] for lattices because one element at a time is added to an already “sorted” sequence.
Table 2 shows the results of performance measurements in the bounded and distributive lattice \((\mathbb{N}, \gcd, \text{lcm}, 1, 0)\). Here, we are using an implementation that is based on the algorithm in Figure (3).

| sequence length | 100  | 1000 | 10000 | 100000 |
|-----------------|------|------|-------|--------|
| time in s       | 0    | 0    | 3.4   | 420    |

Table 2: Wall-clock time for computing \((1, \ldots, n)^\triangle\) according to Equation (8).

These results show that sorting in lattices can now be applied to much larger sequences than those shown in Table (1) before the limitations of an algorithm with quadratic complexity become noticeable.

5 Conclusions

The main results of this paper are Proposition 1 that proves Identity (8) for bounded distributive lattices and Proposition 2 that shows the necessity of the distributivity for Identity (8) to hold.

The remarkable points of Identity (8) are that it exhibits a strong analogy between sorting and Pascal’s triangle, allows to sort in lattices with quadratic complexity, and is in fact a generalization of insertion sort for lattices.

6 Acknowledgment

I am very grateful for the many corrections and valuable suggestions of my colleagues Jochen Burghardt and Hans Werner Pohl.

In particular, Jochen Burghardt’s suggestion to investigate whether the distributivity in Proposition 1 is really necessary led to Proposition 2. Hans Werner Pohl pointed out the analogy of the algorithm in Figure 3 to insertion sort.

References

[1] J. Gerlach. Sorting in Lattices. ArXiv e-prints, March 2013.
[2] S. Roman. Lattices and Ordered Sets. Springer-Verlag New York, 2008.
[3] Donald E. Knuth. The Art of Computer Programming, Volume III: Sorting and Searching. Addison-Wesley, 1973.