Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

An efficient search strategy for hidden ideals in pointed partially ordered sets

Roma Eisel, Valerie McMullen and Robert Muth*

ABSTRACT

We consider a combinatorial question about searching for an unknown ideal μ within a known pointed poset λ. Elements of λ may be queried for membership in μ, but at most k positive queries are permitted. We provide a general search strategy for this problem, and establish new bounds (based on k and the degree and height of λ) for the total number of queries required to identify μ. We show that this strategy performs asymptotically optimally on the family of complete ℓ-ary trees as the height grows.

Keywords: Partially ordered sets, trees, search optimization

2020 Mathematics Subject Classification: 06A07, 68Q25.

1. Introduction

1.1. The hidden ideal problem

Let λ be a finite partially ordered set, and let $k \in \mathbb{N}$. Hidden within λ is a (possibly empty) unknown ideal μ. We seek to identify μ by sequentially querying elements of λ for membership in μ, under the restriction that at most k positive query results are permitted. Letting $\boldsymbol{q}_k(\lambda)$ represent the minimum total number of queries needed to guarantee identification of μ, the *hidden ideal problem* is to solve:

Problem 1.1. *Find $\boldsymbol{q}_k(\lambda)$, and identify a search strategy which realizes this value.*

This problem has been studied under numerous guises and for various families of posets. When λ is totally ordered, Problem 1.1 is related to the 'k-egg' or 'k-marble' problem [8,

* Corresponding author

Received 28 Jul 2025; Revised 28 Sep 2025; Accepted 09 Oct 2025; Published Online 26 Oct 2025.

DOI: 10.61091/jcmcc128-11

12, 14], which is featured in numerous texts on dynamic programming and optimization. When λ is a rectangular poset with one dimension not more than six, Problem 1.1 was settled in [11]. More generally, this problem relates to a broad body of work in computer science and optimization on efficient search within sets with partial order—see for instance [1, 2, 3, 4, 5, 6, 9]—motivated by such applications as debugging, file synchronization, and information retrieval. From this point of view Problem 1.1 is a consideration of this topic under an additional restriction, or a 'cost' imposition on excessive positive queries.

1.2. *A general search strategy*

Contrary to [8, 11, 12, 14], which focus on *optimal* search strategies for *specific* families of posets, our goal in the present paper is to produce *efficient* search strategies for *general* pointed posets (i.e., those with a unique minimal element). Generally speaking, the only known upper bound for $\boldsymbol{q}_k(\lambda)$ is given by $|\lambda|$—a bound achieved via the naive strategy of querying all nodes of λ in a top-down fashion.

In Strategy 3.1 we present a new query strategy which may be applied to any pointed poset λ (in Example 3.1 we provide a short example of Strategy 3.1 at work). We then prove the following main result, which provides a bound on the number of total queries required based on the *degree* and *height* of the poset λ (see Section 2).

Theorem 1.2. *Let $k \in \mathbb{N}$, and let λ be a pointed poset of degree ℓ and height n. Then Strategy 3.1 guarantees identification of the hidden ideal μ, using at most k positive queries, and at most*

$$f_{k,\ell}(n) := \sum_{i=0}^{\lceil n/k\rceil-1} \ell^i + \sum_{j=1}^{k-1} \ell^{\lceil (n-j)/k\rceil} \le \begin{cases} \lceil \frac{n+1}{k} \rceil & \text{if } \ell = 1; \\ k\ell^{\lceil n/k\rceil} & \text{if } \ell > 1, \end{cases}$$

total queries.

This appears as Theorem 4.1 in the body. It follows from Theorem 1.2 then that $\boldsymbol{q}_k(\lambda) \le f_{k,\ell}(n)$ for any pointed poset λ of degree ℓ and height n, yielding a bound that greatly improves on the old bound $\boldsymbol{q}_k(\lambda) \le |\lambda|$ in general. For instance, if λ is the complete ℓ-ary tree of height n (see Section 2.4), then $\boldsymbol{q}_k(\lambda) \le f_{k,\ell}(n) \le k\ell^{(k+1)/k}|\lambda|^{1/k}$.

1.3. *Asymptotic optimality*

It was shown in [11, Proposition 2.2] that $\boldsymbol{q}_k(\lambda)$ has a recursive combinatorial description:

$$\boldsymbol{q}_k(\lambda) = \begin{cases} 0 & \text{if } \lambda = \varnothing; \\ |\lambda| & \text{if } k = 1; \\ \min\{\max\{\boldsymbol{q}_k(\lambda_{\not\succeq u}), \boldsymbol{q}_{k-1}(\lambda_{\succ u})\} \mid u \in \lambda\} + 1 & \text{if } k > 1, \lambda \ne \varnothing. \end{cases}$$

For reasons of storage and computation time, it is quite demanding to compute $\boldsymbol{q}_k(\lambda)$ via this formula for large posets, and optimal search strategies as in Problem 1.1 are linked to the specific structure of λ in very delicate ways (see for instance [11, Algorithm 6.2]). As Strategy 3.1 is a 'one-size-fits-all' approach on the family of pointed posets of a given degree and height, it generally does not produce solutions within an optimally minimal number of total queries, but does in a specific sense perform well on large members of this family, as we now explain.

For $\ell, n \in \mathbb{N}$, let $\mathscr{T}_\ell(n)$ be the complete ℓ-ary tree of height n. Then $\mathscr{T}_\ell(n)$ has maximal cardinality among the family of pointed posets of degree ℓ and height n. Our second main result shows that Strategy 3.1 performs *asymptotically optimally* on complete ℓ-ary trees as the height n increases.

Theorem 1.3. *Fix $k, \ell \in \mathbb{N}$. Then there exist $m, M > 0$ such that*

$$m\ell^{n/k} \leq \boldsymbol{q}_k(\mathscr{T}_\ell(n)) \leq f_{k,\ell}(n) \leq M\ell^{n/k},$$

for all $n \in \mathbb{N}$. Thus $\boldsymbol{q}_k(\mathscr{T}_\ell(n)) = \Theta(\ell^{n/k}) = f_{k,\ell}(n)$ as functions of n.

This appears as Theorem 5.3 in the body. More broadly, this speaks to the overall efficiency of Strategy 3.1. If we fix a poset degree ℓ and a limit k on the number of positive queries, then Theorems 1.2 and 1.3 guarantee that Strategy 3.1 will identify a hidden ideal μ in a pointed poset λ of height n in $\mathscr{O}(\ell^{n/k})$ total queries. On the other hand, Theorem 1.3 shows there always exist such posets (e.g. the complete ℓ-ary trees) which *require* $\Omega(\ell^{n/k})$ total queries to solve.

2. Preliminaries

In this section we briefly lay out preliminaries and relevant definitions. Throughout, we take $\mathbb{N} = \mathbb{Z}_{>0}$, and use the notation $[a, b] := \{a, a+1, \dots, b\}$ for any integers $a \leq b$.

2.1. *Posets and ideals*

We give now a very brief introduction to posets and related terms. The reader should see [7, 10] for a more thorough treatment of the subject.

Definition 2.1. A *partially ordered set*, or *poset*, is a set λ together with a binary relation $\succcurlyeq$, which satisfies the following conditions for all $u, v, w \in \lambda$:
(i) $u \succcurlyeq u$ (reflexivity);
(ii) $u \succcurlyeq v$ and $v \succcurlyeq u$ implies $u = v$ (antisymmetry);
(iii) $u \succcurlyeq v$ and $v \succcurlyeq w$ implies $u \succcurlyeq w$ (transitivity).

From this point forward, we assume all posets under consideration are *finite*. If $u \succcurlyeq v$ then we say that u *dominates* v. For $u, v \in \lambda$, we write $u \succ v$ provided $u \succcurlyeq v$ and $u \neq v$. We say a poset λ is *pointed* provided that for all $a, b \in \lambda$, there exists $c \in \lambda$ with $a \succcurlyeq c$ and $b \succcurlyeq c$. In our context of finite posets, *pointed* implies that either $\lambda = \varnothing$ or λ has a unique minimal element. We will say two posets λ, ν are *isomorphic*, and write $\lambda \cong \nu$, if there exist order-preserving mutually inverse maps $\lambda \rightleftarrows \nu$.

2.1.1. Subposets. If μ is a subset of a poset λ, then μ is itself a poset under the partial order inherited from λ, and we always assume we take this partial order on μ. For $v \in \lambda$, we define some particular subposets of λ:

$$\lambda_{\succcurlyeq v} := \{u \in \lambda \mid u \succcurlyeq v\}, \qquad \lambda_{\succ v} := \{u \in \lambda \mid u \succ v\}, \qquad \lambda_{\not\succcurlyeq v} := \{u \in \lambda \mid u \not\succcurlyeq v\}.$$

More generally, for $S \subseteq \lambda$, we write:

$$\lambda_{\succcurlyeq S} = \bigcup_{v \in S} \lambda_{\succcurlyeq v} = \{u \in \lambda \mid u \succcurlyeq v \text{ for some } v \in S\},$$
$$\lambda_{\not\succcurlyeq S} = \bigcap_{v \in S} \lambda_{\not\succcurlyeq v} = \{u \in \lambda \mid u \not\succcurlyeq v \text{ for all } v \in S\} = \lambda - \lambda_{\succcurlyeq S}.$$

We similarly define $\lambda_{\preccurlyeq v}, \lambda_{\preccurlyeq S}$, and so on. Note that $\lambda_{\succcurlyeq v}$ is always a pointed poset with minimal element v, and $\lambda_{\not\succcurlyeq v}$ is pointed if and only if λ is pointed.

2.1.2. Ideals.

Definition 2.2. Assume $\mu \subseteq \lambda$. We say μ is an *ideal* in λ provided that $\mu = \varnothing$ or $\mu = \lambda_{\preccurlyeq v}$ for some $v \in \lambda$. In the latter case we say μ is *generated* by v.

In our context of finite sets, this notion of ideal is equivalent to the that of a *directed lower set*.

2.1.3. Hasse diagrams. Let λ be a poset, and let $u, v \in \lambda$. We say u *covers* v provided $u \succ v$ and there is no $w \in \lambda$ with $u \succ w \succ v$. We will visually display posets via their *Hasse diagram*—this is a graph whose vertices are the elements of λ, with an edge drawn upward from v to u whenever u covers v.

Example 2.3. In Figure 1 we display the Hasse diagram of a pointed poset λ. The unique minimal element $m \in \lambda$ is shown. The ideal $\lambda_{\preccurlyeq v}$ generated by $v \in \lambda$ is also shown.

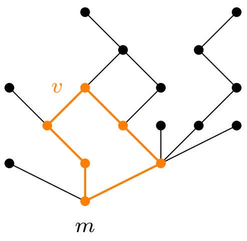


Fig. 1. A pointed poset λ with 17 elements and ideal $\lambda_{\preccurlyeq v}$ highlighted in orange

2.2. Height and degree

Now we introduce some important statistics on posets.

Definition 2.4. Let λ be a poset.

(i) For $v \in \lambda$, we define the *degree* of v to be

$$\deg(v) := \#\{u \in \lambda \mid u \text{ covers } v\}.$$

(ii) We define the *degree* of λ to be

$$\deg(\lambda) := \max\{\deg(v) \mid v \in \lambda\}.$$

Example 2.5. For the poset λ in Figure 1, we have $\deg(v) = 1$, $\deg(m) = 3$, and $\deg(\lambda) = 4$.

A subset C of a pointed poset λ is a *chain* provided every pair of nodes $x, y \in C$ is comparable. We will label the nodes of C as

$$C = \{C_{|C|} \succ C_{|C|-1} \succ \cdots \succ C_1\},$$

where $C_1, \dots, C_{|C|} \in \lambda$. For $v \in \lambda$, we say C is a *v-chain* if $C_{|C|} = v$. We say moreover C is a *maximal* v-chain if $|C| \geq |D|$ for every v-chain D. For a maximal v-chain C, we have that C_t covers C_{t-1} for $t \in [2, |C|]$, and C_1 is the unique minimal element of the pointed poset λ.

Finally, we say a chain C is a *maximal* λ-chain provided that $|C| \geq |D|$ for every chain D in λ.

Definition 2.6. Let λ be a poset.

(i) For $v \in \lambda$, we define the *height* of v to be $\mathrm{ht}(v) := |C|$, where C is any maximal v-chain.

(ii) We define the *height* of λ to be $\mathrm{ht}(\lambda) := |C|$, where C is any maximal λ-chain.

We make a few remarks on height, v-chains and λ-chains that are quickly verified. First, note that $\mathrm{ht}(\lambda) = \max\{\mathrm{ht}(v) \mid v \in \lambda\}$, and $\mathrm{ht}(\lambda_{\preccurlyeq v}) = \mathrm{ht}(v)$ for every $v \in \lambda$. If C is a maximal λ-chain, or maximal v-chain for some $v \in \lambda$, then $\mathrm{ht}(C_t) = t$ for every $t \in [1, |C|]$.

Example 2.7. For the poset λ in Figure 2, we have highlighted two chains. The orange chain is a maximal v-chain. The cyan chain is a maximal w-chain, and a maximal λ-chain. We have

$$\mathrm{ht}(v) = 4, \qquad \mathrm{ht}(w) = 6, \qquad \mathrm{ht}(m) = 1, \qquad \mathrm{ht}(\lambda) = 6.$$

2.3. The posets $\mathscr{P}(\ell, n)$

Going forward, we write

$$\mathscr{P}(\ell, n) = \{\lambda \text{ a pointed poset} \mid \deg(\lambda) \leq \ell,\ \mathrm{ht}(\lambda) \leq n).$$

For example, the poset λ of Figures 1 and 2 is a member of $\mathscr{P}(4, 6)$.

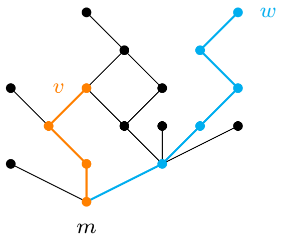


Fig. 2. The pointed poset λ with highlighted chains

Lemma 2.8. *Let $\lambda \in \mathscr{P}(\ell, n)$. Then λ has at most ℓ^{t-1} nodes of height $t \in [1, n]$, and* $|\lambda| \leq \sum_{t=0}^{n-1} \ell^t$.

Proof. For $t \in [1, n]$, write m_t for the number of nodes in λ of height t. Note that since λ is pointed, we must have $m_1 = 1$. Since the degree of λ is ℓ, we also have that $m_t \leq \ell m_{t-1}$ for $t \in [2, n]$. Therefore it follows that $m_t \leq \ell^{t-1}$ for $t \in [1, n]$. Thus

$$|\lambda| = \sum_{t=1}^{n} m_t \leq \sum_{t=1}^{n} \ell^{t-1} = \sum_{t=0}^{n-1} \ell^t, \tag{1}$$

as required. □

2.4. Complete ℓ-ary trees

Definition 2.9. We say a pointed poset $\lambda \in \mathscr{P}(\ell, n)$ is a *complete ℓ-ary tree of height n* provided that λ has exactly ℓ^{t-1} nodes of height $t \in [1, n]$.

Thus the complete ℓ-ary trees of height n are exactly the posets in $\mathscr{P}(\ell, n)$ which render the inequality in Lemma 2.8 an equality. If $\lambda \in \mathscr{P}(\ell, n)$ is a complete ℓ-ary tree of height n, it follows that every node of height less than n has degree ℓ, and $\lambda_{\preccurlyeq v}$ is a chain for every $v \in \lambda$. All complete ℓ-ary trees of height n are isomorphic, and we use the notation $\mathscr{T}_\ell(n)$ to indicate (the isomorphism class of) the complete ℓ-ary tree of height n.

Example 2.10. In Figure 3 we show $\mathscr{T}_5(3)$, the complete 5-ary tree of height 3.

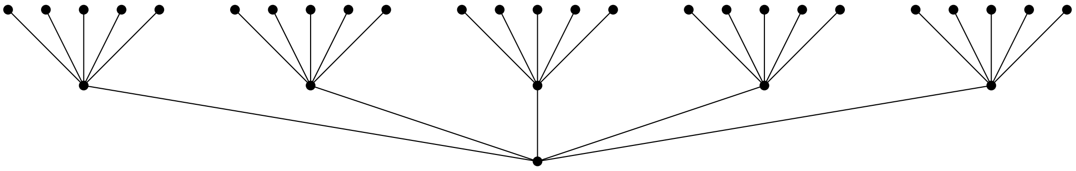


Fig. 3. $\mathscr{T}_5(3)$, the complete 5-ary tree of height 3

2.5. A useful technical lemma on posets

Lemma 2.11. *Let λ be a pointed poset, and assume λ has exactly t nodes $v_1, v_2, \dots, v_t$ of height j. Assume moreover that $w_1, w_2, \dots, w_t \in \lambda$ are such that:*

- $\mathrm{ht}(w_i) \leq j$*, and;*
- w_i *belongs to a maximal* $\lambda_{\not\succcurlyeq\{w_1, w_2, \dots, w_{i-1}\}}$*-chain,*

for all $i \in [1, t]$. Then $\mathrm{ht}(\lambda_{\not\succcurlyeq\{w_1, \dots, w_t\}}) < j$*.*

Proof. We go by induction on t.

Base Case. Assume $t = 1$. Then v_1 is the only node of height j in λ. The node w_1 is in some maximal λ-chain C by assumption, which contains one node of every height $\leq \mathrm{ht}(\lambda)$, and thus contains a node of height j. Thus $v_1 \in C$. Since $\mathrm{ht}(w_1) \leq j$, this implies that $w_1 \preccurlyeq v_1$.

Now assume that $u \in \lambda$ has $\mathrm{ht}(u) \geq j$. Let D be a maximal u-chain in λ. Then since D contains a node of height j as well, and hence contains v_1. Since $\mathrm{ht}(u) \geq \mathrm{ht}(v_1)$, this implies that $u \succcurlyeq v_1 \succcurlyeq w_1$. Therefore $u \notin \lambda_{\not\succcurlyeq w_1}$. Thus $\lambda_{\not\succcurlyeq w_1}$ contains only nodes of λ which have height less than j, so $\mathrm{ht}(\lambda_{\not\succcurlyeq w_1}) < j$, as required.

Induction Step. Now let $t > 1$, and assume the claim holds for any $t' < t$. Again, by assumption, w_1 is in a maximal chain C in λ. This chain must include some height j node v_i. Then, $v_i \succcurlyeq w_1$. Let $S \subseteq [1, t]$ be the set of all s such that $v_s \succcurlyeq w_1$ for all $s \in S$, and $S \neq \varnothing$. Then, the set of vertices of height j in $\lambda_{\not\succcurlyeq\{w_1\}}$ is $\{v_k \mid k \notin S\}$, and there are $t' = t - |S|$ of them. Set $\lambda' = \lambda_{\not\succcurlyeq\{w_1\}}$, and let $v'_1, \dots, v'_{t'}$ be the set of nodes of height j in λ'. Then, $t' < t$. We also set $w'_b = w_{b+1}$ for $b \in [1, t']$. By induction, $\mathrm{ht}(\lambda'_{\not\succcurlyeq\{w'_1, \dots, w'_{t'}\}}) < j$. But we have

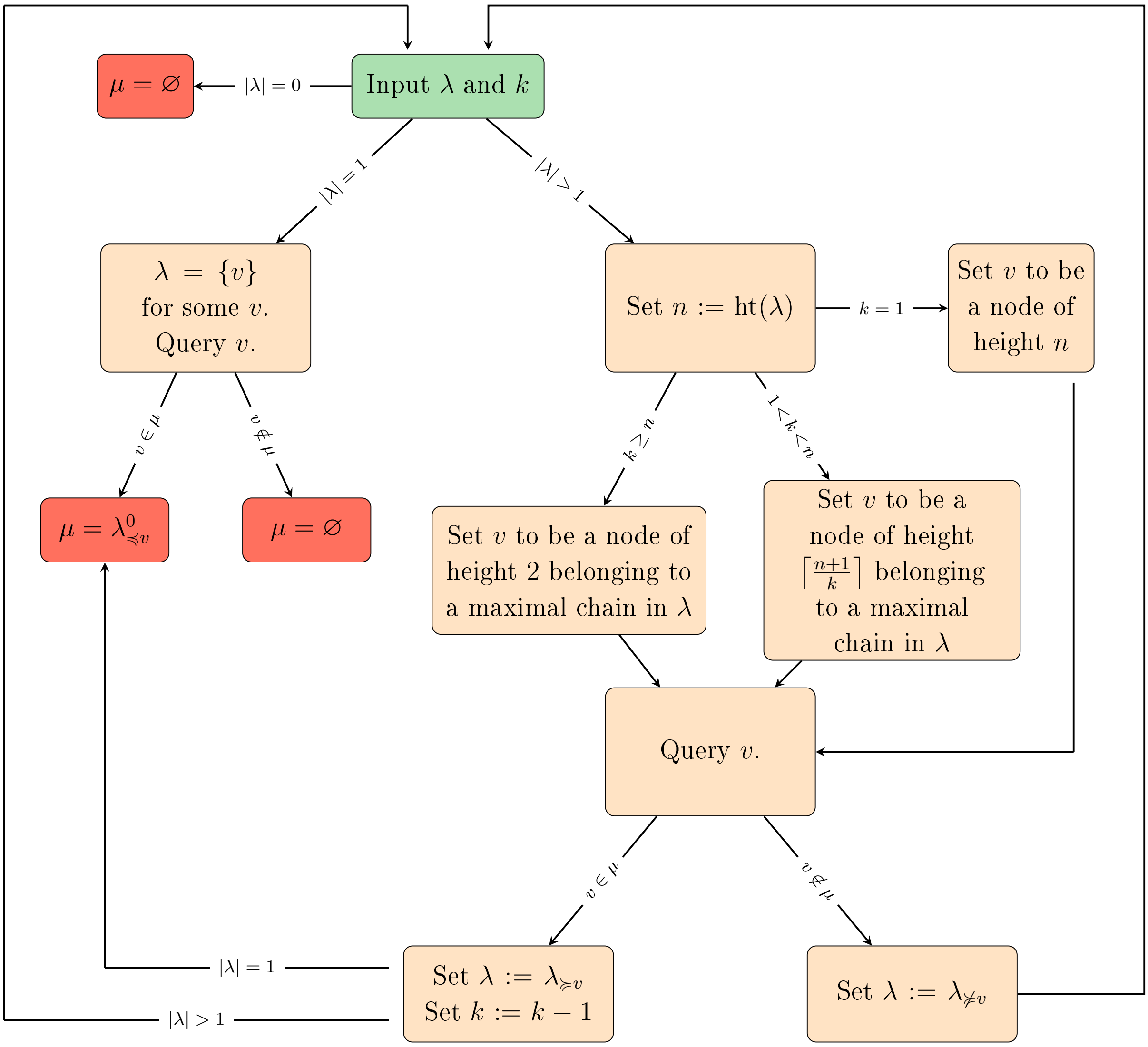
$$\lambda'_{\not\succcurlyeq\{w'_1, \dots, w'_{t'}\}} = (\lambda_{\not\succcurlyeq\{w_1\}})_{\not\succcurlyeq\{w_2, \dots, w_{t'+1}\}} \supseteq (\lambda_{\not\succcurlyeq\{w_1\}})_{\not\succcurlyeq\{w_2, \dots, w_t\}} = \lambda_{\not\succcurlyeq\{w_1, \dots, w_t\}}.$$

Thus, $\mathrm{ht}(\lambda_{\succcurlyeq\{w_1, \dots, w_t\}}) < j$, completing the induction step and the proof. □

3. A general strategy for the hidden ideal problem

Recall now the *hidden ideal problem*, introduced in Section 1.1. We describe in this section a general recursive strategy for identifying a hidden ideal in a pointed poset using a limited amount of positive queries. In Section 4 we establish bounds on the number of total queries this strategy entails, and in Section 5 we examine the asymptotic behavior of this strategy.

Strategy 3.1. *Let λ^0 be a pointed poset with a hidden ideal μ, and let $k^0 \in \mathbb{N}$ be a limit on the number of positive queries permitted. Set $\lambda := \lambda^0$, $k := k^0$, and begin with the green block below. The strategy terminates by identifying μ in one of the red blocks.*


$\mu = \varnothing$
$|\lambda| = 0$
Input λ and k
$|\lambda| = 1$
$|\lambda| > 1$
$\lambda = \{v\}$ for some v. Query v.
Set $n := \mathrm{ht}(\lambda)$
$k = 1$
Set v to be a node of height n
$v \in \mu$
$v \notin \mu$
$k \geq n$
$1 < k < n$
$\mu = \lambda^0_{\preccurlyeq v}$
$\mu = \varnothing$
Set v to be a node of height 2 belonging to a maximal chain in λ
Set v to be a node of height $\lceil \frac{n+1}{k} \rceil$ belonging to a maximal chain in λ
Query v.
$v \in \mu$
$v \notin \mu$
$|\lambda| = 1$
$|\lambda| > 1$
Set $\lambda := \lambda_{\succcurlyeq v}$
Set $k := k - 1$
Set $\lambda := \lambda_{\not\succcurlyeq v}$

Proposition 3.2. *Strategy 3.1 always terminates by correctly identifying the hidden ideal μ in λ^0, while using at most k^0 positive queries.*

Proof. If $\lambda^0 = \varnothing$, then Strategy 3.1 terminates by recognizing $\mu = \varnothing$ and utilizes no queries. If $|\lambda^0| = 1$, then Strategy 3.1 terminates by recognizing either $\mu = \varnothing$ or $\mu = \lambda^0$, and utilizes $1 \leq k^0$ queries. Now set $|\lambda^0| = m > 1$, and assume Strategy 3.1 terminates by identifying any hidden ideal μ^1 in a poset λ^1 utilizing at most k^1 queries whenever $|\lambda^1| < m$. We consider the application of Strategy 3.1 to λ^0.

Case 1. Assume $k = 1$. Then Strategy 3.1 repeatedly queries nodes v of maximal height until either (a) a positive query results, in which case Strategy 3.1 correctly indicates that $\mu = \lambda^0_{\preccurlyeq v}$, or; (b) Strategy 3.1 terminates after (negatively) querying every node in λ^0, and correctly indicates that $\mu = \varnothing$.

Case 2. Assume $k > 1$. Then Strategy 3.1 queries some node $v \in \lambda$ of height greater than 1. We have subcases:

(i) Assume $v \in \mu$. Set $\lambda^1 := \lambda_{\succcurlyeq v}$, $\mu^1 := \mu \cap \lambda_{\succcurlyeq v}$ and $k^1 := k - 1$. Then Strategy 3.1

continues by running the algorithm for (λ^1, k^1), and by induction assumption, terminates by correctly identifying μ^1 as some ideal $\lambda^1_{\succcurlyeq w}$ in λ^1 while using at most k^1 queries. But this implies that $\mu = \lambda^0_{\succcurlyeq w}$, and thus Strategy 3.1 correctly identifies μ using at most $1 + k^1 = k$ positive queries.

(ii) Assume $v \notin \mu$. Set $\lambda^1 := \lambda_{\not\succcurlyeq v}$, $\mu^1 := \mu \cap \lambda_{\not\succcurlyeq v} = \mu$, and $k^1 := k$. Then Strategy 3.1 continues by running the algorithm for (λ^1, k^1), and by induction assumption, terminates by correctly identifying μ^1 as some ideal $\lambda^1_{\succcurlyeq w}$ (or $\varnothing$) in λ^1 while using at most k queries. But this implies that $\mu = \lambda^0_{\succcurlyeq w}$ (or $\varnothing$), and thus Strategy 3.1 correctly identifies μ using at most k positive queries.

Thus in any case, Strategy 3.1 terminates by correctly identifying μ while using at most k positive queries. □

3.1. *Example*

In Figure 4 we display a poset λ^0 of degree 4 and height 6. A hidden ideal μ is shown highlighted in orange. We now show a step-by-step application of Strategy 3.1 which identifies μ while using at most $k^0 = 3$ positive queries.

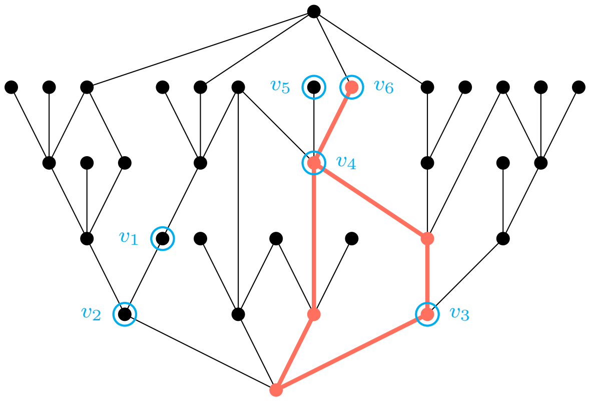


Fig. 4. The poset λ^0 of degree 4 and height 6, with hidden ideal μ highlighted in orange, as described in Example 3.1. The nodes $v_1, \ldots, v_6$ queried during application of Strategy 3.1 are circled in cyan

Query 1: We have $\mathrm{ht}(\lambda) = \mathrm{ht}(\lambda^0) = 6$ and $k = k^0 = 3$, so following Strategy 3.1, we query the node v_1 of height $\lceil \frac{6+1}{3} \rceil = 3$ that resides in a maximal chain in λ. Since $v_1 \notin \mu$, we set $\lambda := \lambda_{\not\succcurlyeq v_1}$, effectively erasing all nodes that dominate v_1.

Query 2: We now have $\mathrm{ht}(\lambda) = 5$ and $k = 3$, so following Strategy 3.1, we query the node v_2 of height $\lceil \frac{5+1}{3} \rceil = 2$ that resides in a maximal chain in λ. Since $v_2 \notin \mu$, we set $\lambda := \lambda_{\not\succcurlyeq v_2}$, effectively erasing all nodes that dominate v_2.

Query 3: We still have $\mathrm{ht}(\lambda) = 5$ and $k = 3$, so following Strategy 3.1, we query the node v_3 of height $\lceil \frac{5+1}{6} \rceil = 2$ that resides in a maximal chain in λ. Since $v_3 \in \mu$, we set $\lambda := \lambda_{\succcurlyeq v_3}$, effectively erasing all nodes that do not dominate v_3. Having used a positive query, we set $k := 3 - 1 = 2$.

Query 4: We now have $\mathrm{ht}(\lambda) = 4$ and $k = 2$, so following Strategy 3.1, we query the node v_4 of height $\lceil \frac{4+1}{2} \rceil = 3$ that resides in a maximal chain in λ. Since $v_4 \in \mu$, we set

$\lambda := \lambda_{\not\succcurlyeq v_4}$, effectively erasing all nodes that do not dominate v_4. Having used a positive query, we set $k := 2 - 1 = 1$.

Query 5: We now have $\mathrm{ht}(\lambda) = 2$ and $k = 1$, so following Strategy 3.1, we query the node v_5 of height 2. Since $v_5 \notin \mu$, we set $\lambda := \lambda_{\not\preccurlyeq v_5}$, effectively erasing v_5.

Query 6: We now have $\mathrm{ht}(\lambda) = 2$ and $k = 1$, so following Strategy 3.1, we query the node v_6 of height 2. Since $v_6 \in \mu$, we have $\mu = \lambda^0_{\preccurlyeq v_6}$, completing the identification strategy.

4. Bounds for Strategy 3.1

In this section we will establish bounds on the maximum number of total queries Strategy 3.1 requires to identify a hidden ideal μ in a poset λ, while using at most k positive queries. Recall from Section 1.2 that we define a function

$$f_{k,\ell}(n) := \sum_{i=0}^{\lceil n/k \rceil - 1} \ell^i + \sum_{j=1}^{k-1} \ell^{\lceil (n-j)/k \rceil},$$

for all $k \in \mathbb{N}$ and $\ell, n \in \mathbb{Z}_{\geq 0}$. We take $0^0 = 1$ when $\ell = 0$. In some special cases, it is straightforward to check that this formula simplifies to:

$$f_{k,\ell}(n) = \begin{cases} k & \text{if } \ell = 0, n = 1; \\ \lceil \frac{n}{k} \rceil + k - 1 & \text{if } \ell = 1; \\ n\ell - \ell + k - n + 1 & \text{if } n \leq k. \end{cases} \tag{2}$$

4.1. *Bounds for Strategy 3.1*

We now establish our first main result. The proof relies on some technical results on ceiling/floor functions and the behavior of the function $f_{k,\ell}(n)$, which we relegate to the Appendix Section 5.

Theorem 4.1. *Let $\ell, n \in \mathbb{Z}_{\geq 0}$, and let $\lambda \in \mathscr{P}(\ell, n)$. Let $\mu \subseteq \lambda$ be a hidden ideal in λ. For $k \in \mathbb{N}$, Strategy 3.1 identifies the hidden ideal μ using at most k positive queries and at most $f_{k,\ell}(n)$ total queries.*

Proof. By Proposition 3.2, we have that Strategy 3.1 identifies μ using at most k positive queries, so we now work on establishing the bound $f_{k,\ell}(n)$ on total queries. We go by induction on $|\lambda|$.

Base Case: Assume $|\lambda|= 1$. Then $\ell = 0, n = 1$. Then Strategy 3.1 identifies μ with one query, and we have $f(0, 1, k) = k \geq 1$, as required.

Induction Step: Assume $\lambda \in \mathscr{P}(\ell, n)$, $|\lambda|> 1$, and the claim holds for all $\lambda' \in \mathscr{P}(\ell', n')$ with $|\lambda'|< |\lambda|$. Say $\lambda \in \mathscr{P}(\ell, n)$. Note that since $|\lambda|> 1$, we have $\ell \geq 1$ and $n \geq 2$. Following Strategy 3.1, we have three cases to consider.

Case 1. Assume $k = 1$. Following Strategy 3.1, we repeatedly query nodes of λ in a top-down fashion until μ is identified upon the first positive query. So at worst, Strategy 3.1

requires $|\lambda|$ queries. But by Lemma 2.8, we have

$$|\lambda|\leq \sum_{i=0}^{n-1} \ell^i = \sum_{i=0}^{\lceil n/1\rceil -1} \ell^i = f_{1,\ell}(n),$$

as required.

Case 2. Assume $k \geq n > 1$. Following Strategy 3.1, we set $v \in \lambda$ to be a node of height 2 in a maximal chain in λ, and query v. We have two possible outcomes:

Case 2.1. Assume $v \notin \mu$. Then $|\lambda_{\not\succeq v}|= 1$, and following Strategy 3.1, we identify μ after one additional query of the base node in λ, for a grand total of two queries. But then by (2) we have

$$f_{k,\ell}(n) = (n-1)\ell + k - n + 1 \geq \ell + 1 \geq 2,$$

as required.

Case 2.2. Assume $v \in \mu$. Then we have used up a positive query, and now have $k-1$ positive queries remaining. Moreover, $\lambda_{\succeq v} \in \mathscr{P}(\ell', n-1)$ for some $\ell' \leq \ell$, and $|\lambda_{\succeq v}|< |\lambda|$. Thus by induction assumption, Strategy 3.1 identifies μ in a total of less than or equal to $f_{k-1,\ell}(n-1)+1$ queries. But by (2) we have

$$f_{k-1,\ell}(n-1)+1 = (n-2)\ell + k - n + 2 \leq (n-1)\ell + k - n + 1 = f_{k,\ell}(n),$$

as required.

Case 3. Assume $n > k > 1$. Set $j = \left\lceil \frac{n+1}{k} \right\rceil$. Assume there are t nodes of height j in $\lambda : v_1, v_2, \ldots, v_t$. Note that t is at most ℓ^{j-1} by Lemma 2.8. We have three possible subcases:

(1) Strategy 3.1 terminates with a solution in less than or equal to t steps.

(2) Strategy 3.1 queries some nodes $w_1, w_2, \ldots, w_{r-1}$, and all return negative results (i.e. not in μ) until w_r returns a positive result, with $r \leq t$.

(3) Strategy 3.1 queries $w_1, w_2, \ldots, w_t$ and all return negative results.

Case 3.1. Assume Strategy 3.1 terminates with a solution in less than or equal to t steps. Then we have

$$t \leq \ell^{j-1} = \ell^{\lceil (n+1)/k\rceil -1} = \ell^{\lceil n-(k-1)/k\rceil} \leq \sum_{i=0}^{\lceil n/k\rceil -1} \ell^i + \sum_{t=1}^{k-1} \ell^{\lceil (n-t)/k\rceil} = f_{k,\ell}(n),$$

as required.

Case 3.2. Assume Strategy 3.1 queries nodes $w_1, w_2, \ldots, w_{r-1}$, all with negative results (i.e. not in μ) until w_r returns with a positive result, with $r \leq t$. Set $n_0 = n$. Set $n_i = \mathrm{ht}(\lambda_{\not\succeq \{w_1,\ldots,w_i\}})$ for $i = 1, \ldots, r-1$. It follows from Strategy 3.1 and the above assumptions that $\mathrm{ht}(w_i) = \left\lceil \frac{n_{i-1}+1}{k} \right\rceil$. Then, $\mathrm{ht}(w_r) = \left\lceil \frac{n_{r-1}+1}{k} \right\rceil$. Also, note $n = n_0 \geq n_1 \geq \cdots \geq n_{r-1}$. Since w_r is a height $\left\lceil \frac{n_{r-1}+1}{k} \right\rceil$ node in a maximal chain in $\lambda_{\not\succeq \{w_1,\ldots,w_{r-1}\}}$, we have

$$\begin{aligned}\mathrm{ht}((\lambda_{\not\succeq \{w_1,\ldots,w_{r-1}\}})_{\succeq w_r}) &= n_{r-1} - \left\lceil \frac{n_{r-1}+1}{k} \right\rceil + 1 = \left\lfloor \frac{(n_{r-1}+1)(k-1)}{k} \right\rfloor \\ &\leq \left\lfloor \frac{(n+1)(k-1)}{k} \right\rfloor = n - \left\lceil \frac{n+1}{k} \right\rceil + 1,\end{aligned}$$

where the second and third equalities follow from Lemma 5.4. Set $\lambda' = (\lambda_{\not\geq\{w_1,\ldots,w_{r-1}\}})_{\geqslant w_r}$, and $n' = \operatorname{ht}\left((\lambda_{\not\geq\{w_1,\ldots,w_{r-1}\}})_{\geqslant w_r}\right)$. Since $|\lambda'|< |\lambda|$, and $\lambda' \in \mathscr{P}(\ell', n')$, we have by induction assumption that Strategy 3.1 solves λ' can in less than or equal to $f_{k-1,\ell}(n')$ queries. Thus Strategy 3.1 solves λ in less than or equal to $r + f_{k-1,\ell}(n')$ queries. We have

$$\begin{aligned} r + f_{k-1,\ell}(n') &\leq \ell^{\left\lceil \frac{n+1}{k} \right\rceil -1} + \sum_{i=0}^{\left\lceil \frac{n-\left\lceil \frac{n+1}{k}\right\rceil +1}{k-1} \right\rceil -1} \ell^i + \sum_{t=1}^{k-2} \ell^{\left\lceil \frac{n-\left\lceil \frac{n+1}{k}\right\rceil +1-t}{k-1} \right\rceil} \\ &= \ell^{\left\lceil \frac{n-(k-1)}{k} \right\rceil} + \sum_{i=0}^{\left\lceil \frac{\left\lceil \frac{n(k-1)}{k}\right\rceil}{k-1} \right\rceil -1} \ell^i + \sum_{t=1}^{k-2} \ell^{\left\lceil \frac{\left\lceil \frac{n(k-1)}{k}\right\rceil -t}{k-1} \right\rceil} \\ &= \ell^{\left\lceil \frac{n-(k-1)}{k} \right\rceil} + \sum_{i=0}^{\left\lceil \frac{n}{k} \right\rceil -1} \ell^i + \sum_{t=1}^{k-2} \ell^{\left\lceil \frac{n-t}{k} \right\rceil} \\ &= f_{k,\ell}(n), \end{aligned}$$

where the first equality follows by Lemma 5.5, and the second equality follows by Lemma 5.6.

Case 3.3. Assume the algorithm queries $w_1, \ldots, w_t$, and all return negative results. Then, note that $\operatorname{ht}(w_i) \leq j$ for all $i \in [1, t]$ and, by construction of the algorithm, each w_i belongs to a maximal chain in $\lambda_{\not\geq\{w_1,\ldots,w_{i-1}\}}$. Thus, by Lemma 2.11, we have $\operatorname{ht}(\lambda_{\not\geq\{w_1,\ldots,w_t\}}) < j$. So, after querying $w_1, \ldots, w_t$, we continue by applying Strategy 3.1 to $\lambda_{\not\geq\{w_1,\ldots,w_t\}}$. By induction assumption, Strategy 3.1 solves $\lambda_{\not\geq\{w_1,\ldots,w_t\}}$ in less than or equal to $f_{k,\ell}(j-1)$ queries, and thus solves λ in less than or equal to $t+f_{k,\ell}(j-1)$ queries. By Lemma 5.7, we have

$$t + f_{k,\ell}(j-1) \leq \ell^{j-1} + f_{k,\ell}(j-1) \leq f_{k,\ell}(n),$$

which completes the induction step and the proof. □

Now recall from Section 1.1 the value $\boldsymbol{q}_k(\lambda)$ which records the *minimal* necessary queries to identify a hidden ideal μ in a poset λ granted at most k positive queries.

Corollary 4.2. *Let $k \in \mathbb{N}$, and let λ be a pointed poset of degree ℓ and height n. Then we have*

$$\boldsymbol{q}_k(\lambda) \leq f_{k,\ell}(n) = \sum_{i=0}^{\lceil n/k\rceil -1} \ell^i + \sum_{j=1}^{k-1} \ell^{\lceil (n-j)/k\rceil} \leq (k\ell)\ell^{n/k}.$$

Proof. The first inequality follows from Theorem 4.1, so we focus now on the second inequality. If $\ell = 1$, this is immediate from (2). If $\ell \geq 2$ we have

$$\begin{aligned} f_{k,\ell}(n) &= \sum_{i=0}^{\lceil n/k \rceil - 1} \ell^i + \sum_{j=1}^{k-1} \ell^{\lceil (n-j)/k \rceil} = \frac{\ell^{\lceil n/k \rceil} - 1}{\ell - 1} + \sum_{j=1}^{k-1} \ell^{\lceil (n-j)/k \rceil} \\ &\leq \ell^{\lceil n/k \rceil} + \sum_{j=1}^{k-1} \ell^{\lceil n/k \rceil} = k\ell^{\lceil n/k \rceil} \leq k\ell^{1+n/k} = (k\ell)\ell^{n/k}, \end{aligned}$$

as required. □

5. Asymptotic optimality of Strategy 3.1

We first recall here some results from [11] which will be useful. For any $x \in \mathbb{Z}_{\geq 0}$, let $T_k(x) := \sum_{i=1}^{k} \binom{x}{k}$, and define $\tau_k(x)$ to be the smallest integer such that $x \leq T_k(\tau_k(x))$. The following result from [11, Theorem 4.2] gives lower bounds on the value of $\boldsymbol{q}_k(\lambda)$.

Theorem 5.1. *For all $k \in \mathbb{N}$ and posets λ, we have $\tau_k(|\lambda|) \leq \boldsymbol{q}_k(\lambda)$.*

Recall that $\mathscr{T}_\ell(n)$ is the complete ℓ-ary tree of height n.

Lemma 5.2. *Fix $k, \ell \in \mathbb{N}$. There exists $m > 0$ such that for all $n \in \mathbb{N}$, we have $m\ell^{n/k} \leq \boldsymbol{q}_k(\mathscr{T}_\ell(n))$.*

Proof. First, note that

$$T_k(x) = \sum_{i=1}^{k} \binom{x}{k} = \sum_{i=1}^{k} \frac{1}{k!} x(x-1)\cdots(x-k+1),$$

is a degree k polynomial in x, and that by definition $T_k(x)$ is an increasing function on x. Thus we may write $T_k(x) = \sum_{i=0}^{k} a_i x^i$, for some coefficients $a_0, \ldots, a_k \in \mathbb{R}$. Set $u = |a_0| + \cdots + |a_k|$. Then for all $x \in \mathbb{N}$ we have

$$T_k(x) = \sum_{i=0}^{k} a_i x^i \leq \sum_{i=0}^{k} |a_i| x^i \leq \sum_{i=0}^{k} |a_i| x^k = ux^k.$$

Now define a function $\Gamma_k(x) = \left(\frac{x}{u}\right)^{1/k}$, and set $m = (\ell u)^{-1/k}$. Since $T_k(x) \leq ux^k$ and T_k is increasing in x, we have

$$T_k(\Gamma_k(x)) \leq \Gamma_k(ux^k) = \left(\frac{ux^k}{u}\right)^{1/k} = x \leq T_k(\tau_k(x)),$$

which again implies that $\Gamma_k(x) \leq \tau_k(x)$ since T_k is increasing. Then for $n \in \mathbb{N}$ we have

$$\begin{aligned} m\ell^{n/k} &= \frac{\ell^{n/k}}{(\ell u)^{1/k}} = \frac{\ell^{(n-1)/k}}{u^{1/k}} = \left(\frac{\ell^{n-1}}{u}\right)^{1/k} = \Gamma_k(\ell^{n-1}) \\ &\leq \Gamma_k\left(\sum_{i=0}^{n-1} \ell^i\right) = \Gamma_k(|\mathscr{T}_\ell(n)|) \leq \tau_k(|\mathscr{T}_\ell(n)|) \leq \boldsymbol{q}_k(\mathscr{T}_\ell(n)), \end{aligned}$$

where we have used the fact that Γ_k is increasing for the first inequality, and Theorem 5.1 for the last inequality. □

Now we are finally in position to prove our second main result, which shows that Strategy 3.1 performs asymptotically optimally on the family of ℓ-ary trees as the height n grows.

Theorem 5.3. *Fix $k, \ell \in \mathbb{N}$. Then there exist $m, M > 0$ such that*

$$m\ell^{n/k} \leq \boldsymbol{q}_k(\mathscr{T}_\ell(n)) \leq f_{k,\ell}(n) \leq M\ell^{n/k},$$

for all $n \in \mathbb{N}$. Thus $\boldsymbol{q}_k(\mathscr{T}_\ell(n)) = \Theta(\ell^{n/k}) = f_{k,\ell}(n)$ as functions of n.

Proof. By Lemma 5.2 there exists $m > 0$ such that $m\ell^{n/k} \leq \boldsymbol{q}_k(\mathscr{T}_\ell(n))$. Taking $M = k\ell$, we have by Corollary 4.2 that $\boldsymbol{q}_k(\mathscr{T}_\ell(n)) \leq f_{k,\ell}(n) \leq M\ell^{n/k}$ since $\mathscr{T}_\ell(n) \in \mathscr{P}(\ell, n)$, completing the proof. □

Acknowledgements

This paper was initially motivated by a problem introduced by Jonathan Kujawa in [13], which recast the search for pollution sources in the Mississippi river system as a hidden ideal problem in a tree poset. We thank him for the inspiration.

References

[1] Y. Ben-Asher, E. Farchi, and I. Newman. Optimal search in trees. *SIAM Journal on Computing*, 28(6):2090–2102, 1999. https://doi.org/10.1137/S009753979731858X.

[2] Y. Chen. An efficient search algorithm for partially ordered sets. In volume 2006, pages 91–94, 2006.

[3] F. Cicalese, T. Jacobs, E. Laber, and M. Molinaro. On greedy algorithms for decision trees. In volume 2010, pages 206–217, 2010. https://doi.org/10.1007/978-3-642-17514-5_18.

[4] F. Cicalese, T. Jacobs, E. Laber, and M. Molinaro. On the complexity of searching in trees and partially ordered structures. *Theoretical Computer Science*, 412:6879–6896, 2011. https://doi.org/10.1016/j.tcs.2011.08.042.

[5] F. Cicalese, B. Keszegh, B. Lidický, D. Pálvölgyi, and T. Valla. On the tree search problem with non-uniform costs. *Theoretical Computer Science*, 647:22–32, 2016. https://doi.org/10.1016/j.tcs.2016.07.019.

[6] C. Daskalakis, R. M. Karp, E. Mossel, S. J. Riesenfeld, and E. Verbin. Sorting and selection in posets. *SIAM Journal on Computing*, 40(3):597–622, 2011. https://doi.org/10.1137/070697720.

[7] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, New York, 2002.

[8] R. Denman, D. Hailey, and M. Rothenberg. The tower and glass marbles problem. *College Mathematics Journal*, 41(5):350–356, 2010. https://doi.org/10.4169/074683410X521946.

[9] D. Dereniowski. Edge ranking and searching in partial orders. *Discrete Applied Mathematics*, 156(13):2493–2500, 2008. https://doi.org/10.1016/j.dam.2008.03.007.

[10] B. Dushnik and E. Miller. Partially ordered sets. *American Journal of Mathematics*, 63, 1941. https://doi.org/10.2307/2371374.

[11] A. Iams, H. Johnston, and R. Muth. Searching for quicksand ideals in partially ordered sets. *Australasian Journal of Combinatorics*, 79(2):256–283, 2021.

[12] J. D. E. Konhauser, D. Velleman, and S. Wagon. Which way did the bicycle go? *Dolciani Mathematical Expositions, The Mathematical Association of America*, 18, 1996.

[13] J. Kujawa. Sleepy watchmen, meandering rivers, and cuspidal ribbons, 2021. https://3quarksdaily.com/3quarksdaily/2020/09/sleepy-watchmen-meandering-rivers-and-cuspidal-ribbons.html.

[14] S. Skiena. *The Algorithm Design Manual*. Springer-Verlag, New York, 1997.

Roma Eisel
Department of Mathematics and Computer Science, Duquesne University
Pittsburgh PA, USA 15282
E-mail eiselr@duq.edu

Valerie McMullen
Department of Mathematics and Computer Science, Duquesne University
Pittsburgh PA, USA 15282
E-mail mcmullenv@duq.edu

Robert Muth
Department of Mathematics and Computer Science, Duquesne University
Pittsburgh PA, USA 15282
E-mail muthr@duq.edu

Appendix

In this section, we will establish the results on the function $f_{k,\ell}(n)$ and ceiling/floor functions that are used in our proof of Theorem 4.1.

Lemma 5.4. *Let $m, k \in \mathbb{N}$. Then we have $m - \lceil \frac{m+1}{k} \rceil + 1 = \lfloor \frac{(m+1)(k-1)}{k} \rfloor$.*

Proof. We have

$$\begin{aligned} m-\left\lceil\frac{m+1}{k}\right\rceil+1 &= -\left(-m+\left\lceil\frac{m+1}{k}\right\rceil-1\right) = -\left\lceil -m+\frac{m+1}{k}-1\right\rceil \\ &= -\left\lceil\frac{(m+1)(1-k)}{k}\right\rceil = \left\lfloor\frac{(m+1)(k-1)}{k}\right\rfloor, \end{aligned}$$

as required. □

Lemma 5.5. *Let $n, k \in \mathbb{N}$. Then we have*

$$n-\left\lceil\frac{(k-1)n}{k}\right\rceil = \left\lceil\frac{n-k+1}{k}\right\rceil = \left\lceil\frac{n+1}{k}\right\rceil - 1.$$

Proof. The second equality is obvious. We prove the first equality. By the division algorithm we have $n = kq + r$ for some $r \in [0, k-1]$ and $q \in \mathbb{Z}$. Then we have

$$\begin{aligned} n-\left\lceil\frac{(k-1)n}{k}\right\rceil &= (kq+r)-\left\lceil\frac{(k-1)(kq+r)}{k}\right\rceil = (kq+r)-\left\lceil\frac{k^2q+kr-kq-r}{k}\right\rceil \\ &= kq+r-\left\lceil kq+r-q-\frac{r}{k}\right\rceil = kq+r-kq-r+q+\left\lceil\frac{-r}{k}\right\rceil \\ &= q+\left\lceil\frac{-r}{k}\right\rceil = q. \end{aligned}$$

On the other hand, we have

$$\begin{aligned} \left\lceil\frac{n-k+1}{k}\right\rceil &= \left\lceil\frac{(kq+r)-(k-1)}{k}\right\rceil = \left\lceil\frac{kq-k+r+1}{k}\right\rceil \\ &= \left\lceil(q-1)+\frac{r+1}{k}\right\rceil = q-1+\left\lceil\frac{r+1}{k}\right\rceil = q-1+1 = q. \end{aligned}$$

Therefore, $n-\left\lceil\frac{(k-1)n}{k}\right\rceil = \left\lceil\frac{n-(k-1)}{k}\right\rceil$, as required. □

Lemma 5.6. *Let $n, k \in \mathbb{N}$, $k \geq 2$, and let $i \in [0, k-2]$. Then we have*

$$\left\lceil\frac{\left\lceil\frac{nk-n}{k}\right\rceil - i}{k-1}\right\rceil = \left\lceil\frac{n-i}{k}\right\rceil.$$

Proof. Write $n = kq + r$ for some $r \in [0, k-1]$ and $q \in \mathbb{Z}$. Then we have

$$\begin{aligned}
\left\lceil \frac{\left\lceil \frac{nk-n}{k} \right\rceil - i}{k-1} \right\rceil &= \left\lceil \frac{\left\lceil \frac{(k-1)(kq+r)}{k} \right\rceil - i}{k-1} \right\rceil = \left\lceil \frac{\left\lceil \frac{k^2q+kr-kq-r}{k} \right\rceil - i}{k-1} \right\rceil \\
&= \left\lceil \frac{(kq+r-q+\lceil\frac{-r}{k}\rceil) - i}{k-1} \right\rceil = \left\lceil \frac{(kq+r-q+0) - i}{k-1} \right\rceil \\
&= \left\lceil q + \frac{r-i}{k-1} \right\rceil = q + \left\lceil \frac{r-i}{k-1} \right\rceil \\
&= \begin{cases} q & \text{if } r \leq i, \\ q+1 & \text{if } r > i. \end{cases}
\end{aligned}$$

On the other hand, we have

$$\left\lceil \frac{n-i}{k} \right\rceil = \left\lceil \frac{kq+r-i}{k} \right\rceil = \left\lceil q + \frac{r-i}{k} \right\rceil = q + \left\lceil \frac{r-i}{k} \right\rceil = \begin{cases} q & \text{if } r \leq i, \\ q+1 & \text{if } r > i. \end{cases}$$

Therefore, $\left\lceil \frac{\lceil\frac{(k-1)n}{k}\rceil - i}{k-1} \right\rceil = \lceil \frac{n-i}{k} \rceil$, as required. □

Lemma 5.7. *Assume $n > k \geq 2$, $\ell \geq 1$, and set $j = \lceil \frac{n+1}{k} \rceil$. Then we have:*

$$\ell^{j-1} + f_{k,\ell}(j-1) \leq f_{k,\ell}(n).$$

Proof. We first consider the case $\ell = 1$. Note that $n+1 \geq k+2$, and so

$$k^2 \leq k^2 + k - 2 = (k+2)(k-1) \leq (n+1)(k-1) = nk - n + k - 1.$$

Therefore we have $k^2 - nk + n - k + 1 \leq 0$, and so

$$0 \geq \left\lceil \frac{k^2 - nk + n - k + 1}{k} \right\rceil = \left\lceil \frac{n+1}{k} \right\rceil - 1 + k - n = j - 1 + k - n.$$

Thus $k + j - 1 \leq n$, and so

$$1 + \left\lceil \frac{j-1}{k} \right\rceil = \left\lceil \frac{k+j-1}{k} \right\rceil \leq \left\lceil \frac{n}{k} \right\rceil,$$

which implies in view of (2) that

$$1^{j-1} + f_{k,1}(j-1) = 1 + \left\lceil \frac{j-1}{k} \right\rceil + k - 1 \leq \left\lceil \frac{n}{k} \right\rceil + k - 1 = f_{k,1}(n),$$

completing the proof in the case $\ell = 1$. From now on we assume $\ell \geq 2$.

Case 1. Assume $\lceil \frac{n}{k} \rceil = \lceil \frac{n+1}{k} \rceil = j$. Note that since that $(k-1)(j-1) \geq 0$, we have $-1 \geq \frac{j-jk-1}{k}$, and thus $0 > \lceil \frac{j-jk-1}{k} \rceil = \lceil \frac{j-1}{k} \rceil - j$. Therefore $j - 2 \geq \lceil \frac{j-1}{k} \rceil - 1$. Note that since $n \geq \lceil \frac{n}{k} \rceil = j > j-1$, we have $\lceil \frac{n-u}{k} \rceil \geq \lceil \frac{j-1-u}{k} \rceil$ for all $u \in [1, k-1]$. Therefore

$$\begin{aligned} f_{k,\ell}(n) &= \ell^{j-1} + \sum_{t=0}^{j-2} \ell^t + \sum_{u=1}^{k-1} \ell^{\lceil (n-u)/k \rceil} \geq \ell^{j-1} + \sum_{t=0}^{\lceil (j-1)/k \rceil - 1} \ell^t + \sum_{u=1}^{k-1} \ell^{\lceil (j-1-u)/k \rceil} \\ &= \ell^{j-1} + f_{k,\ell}(j-1), \end{aligned}$$

as required.

Case 2. Assume $k = 2$, and $\lceil \frac{n}{2} \rceil < \lceil \frac{n+1}{2} \rceil = \lceil \frac{n}{2} \rceil + 1$. It follows then that $n \equiv 0 \pmod 4$ or $n \equiv 2 \pmod 4$. We approach these subcases separately.

Case 2.1. Assume $n \equiv 0 \pmod 4$. Then $n = 4m$ for some $m \geq 1$. Then $j = \lceil \frac{4m+1}{2} \rceil = 2m + 1$. Then we have

$$f(\ell, n, k) = \sum_{t=0}^{2m-1} \ell^t + \ell^{2m} \geq \ell^{2m} + \sum_{t=0}^{m} \ell^t = \ell^{j-1} + f(\ell, j-1, k),$$

as required.

Case 2.2. Assume $n \equiv 2 \pmod 4$. Then $n = 4m+2$ for some $m \geq 1$, and $j = \lceil \frac{4m+3}{2} \rceil = 2m + 2$. Then we have

$$f(\ell, n, k) = \sum_{t=0}^{2m} \ell^t + \ell^{2m+1} \geq \ell^{2m+1} + \sum_{t=0}^{m} \ell^t + \ell^m = \ell^{j-1} + f(\ell, j-1, k),$$

as required.

Case 3. Finally, assume $k \geq 3$ and $\lceil \frac{n}{k} \rceil < \lceil \frac{n+1}{k} \rceil = \lceil \frac{n}{k} \rceil + 1$. Then we have $n = km$ and $j = m + 1$ for some $m \geq 2$. Since $(k-1)m \geq k - 1$, we have $m \geq \frac{m+k-1}{k}$. Therefore we have

$$\begin{aligned} \ell^{\lceil (n-1)/k \rceil} &= \ell^{\lceil m - 1/k \rceil} = \ell^m = \ell^{\lceil m \rceil} \geq \ell^{\lceil (m+k-1)/k \rceil} = \ell^{\lceil (m-1)/k \rceil + 1} = \ell \cdot \ell^{\lceil (m-1)/k \rceil} \\ &\geq 2\ell^{\lceil (m-1)/k \rceil} = \ell^{\lceil (m-1)/k \rceil} + \ell^{\lceil (m-1)/k \rceil} \geq \ell^{\lceil (m-1)/k \rceil} + \ell^{\lceil (m-2)/k \rceil}. \end{aligned}$$

and

$$\ell^{\lceil (n-2)/k \rceil} = \ell^{\lceil m - 2/k \rceil} = \ell^m = \ell^{j-1}.$$

Thus we have

$$\begin{aligned} f_{k,\ell}(n) &= \sum_{i=0}^{m-1} \ell^i + \ell^{\lceil (n-1)/k \rceil} + \ell^{\lceil (n-2)/k \rceil} + \sum_{t=3}^{k-1} \ell^{\lceil (n-t)/k \rceil} \\ &\geq \sum_{i=0}^{\lceil \frac{m}{k} \rceil - 1} \ell^i + \left(\ell^{\lceil (m-1)/k \rceil} + \ell^{\lceil (m-2)/k \rceil} \right) + \ell^{j-1} + \sum_{t=3}^{k-1} \ell^{\lceil (m-t)/k \rceil} \\ &= \ell^{j-1} + f_{k,\ell}(j-1), \end{aligned}$$

as required, completing the proof. □