AUSTRALASIAN JOURNAL OF COMBINATORICS
Volume **79(2)** (2021), Pages 256–283

# Searching for quicksand ideals in partially ordered sets

Alexas Iams Hannah Johnston Robert Muth

*Department of Mathematics*
*Washington & Jefferson College*
*Washington, PA 15301*
*U.S.A.*

iamsal@washjeff.edu johnstonhm@washjeff.edu rmuth@washjeff.edu

**Abstract**

We consider a combinatorial question about searching for an unknown ideal $\mu$ within a known poset $\lambda$. Elements of $\lambda$ may be queried for membership in $\mu$, but at most $k$ positive query results are permitted. The goal is to find a search strategy which guarantees a solution in a minimal total number $\mathsf{q}_k(\lambda)$ of queries. We provide tight bounds for $\mathsf{q}_k(\lambda)$, and construct optimal search strategies for the case where $k = 2$ and $\lambda$ is the product poset of totally ordered finite sets, one of which has cardinality not more than six.

## 1 Introduction

### 1.1 Quicksand puzzle

A surveyor stands in the northeast corner of a rectangular field $\lambda$ of dimension $m \times n$. In the southwest corner of the field there may exist a rectangular quicksand pit $\mu$ of unknown dimension $m' \times n'$. The surveyor has $k$ stones available to toss into the field in order to identify safe and unsafe regions of the field.

In order to gain information, the surveyor tosses a stone into some location $x$ in the field. If the stone does not sink, it follows that the region northeast of $x$ is safe; the surveyor can venture into the field to retrieve the stone and use it again. If the stone does sink, the surveyor knows that the quicksand pit extends at least as far as $x$, but they now have one less stone with which to work (see Figure 1). How can the surveyor identify the location of the quicksand pit, and do so in a minimal number of tosses?

ISSN: 2202-3518 

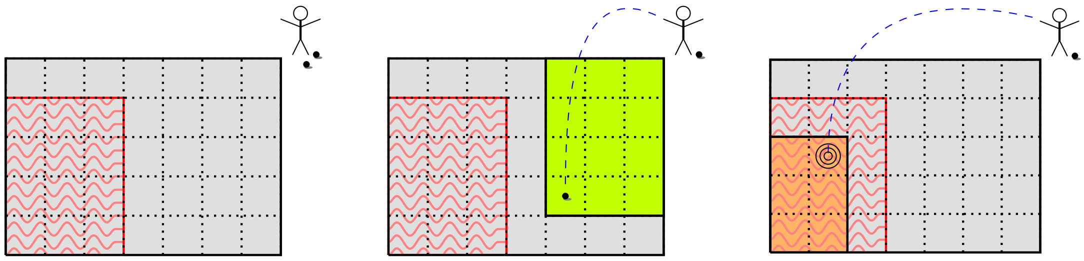



**Figure 1:** Surveying a $5 \times 7$ field with hidden $4 \times 3$ quicksand pit.

### 1.2 Quicksand ideals in posets

As we explain in Section 1.3, this puzzle is a special case of a more general problem. Let $\lambda$ be a finite poset and $k \in \mathbb{N}$. We seek to identify a (possibly empty) 'quicksand' ideal $\mu$ contained in $\lambda$ by sequentially querying elements of $\lambda$ for membership in $\mu$, under the restriction that at most $k$ positive query results are permitted. Letting $\mathsf{q}_k(\lambda)$ represent the minimum total number of queries needed to guarantee identification of $\mu$, our goal is to solve:

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

For all $k \in \mathbb{N}$, the value $\mathsf{q}_k(\lambda)$ has a recursive combinatorial description, as shown in Proposition 2.2:

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

For any $x \in \mathbb{Z}_{\geq 0}$, let $T_k(x) = \sum_{i=1}^{k} \binom{x}{k}$, and let $\tau_k(x)$ be the smallest integer such that $x \leq T_k(\tau_k(x))$. Our first main result provides bounds for $\mathsf{q}_k(\lambda)$:

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

This appears as Theorem 4.2 in the text. These bounds are tight, in that $\mathsf{q}_k(\lambda) = |\lambda|$ when $\lambda$ has the trivial partial order, and $\mathsf{q}_k(\lambda) = \tau_k(|\lambda|)$ when $\lambda$ is totally ordered. In fact, when $\lambda$ is totally ordered, Problem 1.1 is related to the '$k$-egg' or '$k$-marble' problem [**9**, **12**–**14**], which appears in numerous texts on dynamic programming and optimization, and perhaps apocryphally, as an interview question for certain coding positions in big tech.

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 [**2**–**5**, **7**, **10**]—motivated by such applications as debugging, file synchronization, and information retrieval. From this point of view, our Problem 1.1 is a consideration of this topic under an additional restriction on queries. One may view this restriction from a cost-minimization perspective, in which the limit on positive queries in this paper corresponds to the $(k+1)$st positive query having large cost. Searches in posets with non-uniform costs were studied in [**6**] (where cost is tied to querying particular nodes), and in [**1**] (where query cost is randomized).

### 1.3 Quicksand ideals in the product order, $k = 2$ case

After investigating general results described in Section 1.2, we devote our attention to a special case of Problem 1. When $\kappa, \nu$ are totally ordered sets, we consider $\kappa \times \nu$ to be a poset under the *product partial order*; i.e.,

$$(x_1, y_1) \succeq (x_2, y_2) \iff x_1 \geq x_2 \text{ and } y_1 \geq y_2,$$

for $x_1, x_2 \in \kappa$ and $y_1, y_2 \in \nu$. We consider the $k = 2$ case, where $T_2(x)$ is the triangular number $1 + 2 + \cdots + x = x(x+1)/2$, and $\tau_2(x) = \lceil(\sqrt{8x+1} - 1)/2\rceil$. Our second main result, which appears as Corollary 6.2 in the text, provides a partial solution to Problem 1.1 in this setting:

**Theorem B.** *Let $\kappa, \nu$ be finite totally ordered sets, with $|\kappa| \leq 6$ or $|\nu| \leq 6$. Then*

$$\mathsf{q}_2(\kappa \times \nu) = \begin{cases} 9 & \text{if } |\kappa| = |\nu| = 6; \\ \tau_2(|\kappa||\nu|) & \text{otherwise.} \end{cases}$$

In Algorithm 6.2 we describe an explicit strategy, for any such $\kappa, \nu$, which realizes the value $\mathsf{q}_2(\kappa \times \nu)$ above. In general, this strategy—and hence the proof of Theorem B—is rather delicately connected to the congruence class of $\tau_2(|\kappa||\nu|)$ modulo $|\kappa|$ and $|\nu|$, and relies heavily on some interesting number theoretic facts about triangular numbers proved in Section 3.2. We close the paper with a conjectural upper bound on $\mathsf{q}_2(\kappa \times \nu)$ in general, see Section 6.1.

### 1.4 Solving the quicksand puzzle

Theorem B offers a solution to the puzzle in Section 1.1 for the case where $k = 2$ and one dimension of the field is not more than six. Indeed, we may consider the field $\lambda$ as the poset $[1, m] \times [1, n]$, depicted as a rectangular array of boxes in the first quadrant of the Cartesian plane. The quicksand pit is then an unknown ideal in $\lambda$, since any ideal $\mu \subseteq \lambda$ is either empty or equal to $[1, m'] \times [1, n']$ for some $m' \leq m$, $n' \leq n$.

Take the $k = 2$, $\lambda = [1, 5] \times [1, 7]$ example from Section 1.1 for instance. Algorithm 6.2 returns an optimal strategy displayed in Figure 2. The surveyor tosses their first stone into the locations marked ①, ②, ③, . . . , in sequence. If this stone never sinks, then $\mu = \varnothing$. If the stone sinks on say, the $i$th toss, the remaining uncleared area weakly northeast of this location (belonging to the same colored region as ⓘ), is checked sequentially with the remaining stone, beginning with northeastern-most nodes and working (weakly) to the southwest. When the second stone sinks it will determine the northeast corner of $\mu$, and if it never sinks, the northeast corner of $\mu$ is at $\textcircled{i}$. This strategy identifies the quicksand pit $\mu$ in at most $\tau_2(5 \cdot 7) = 8$ total tosses.

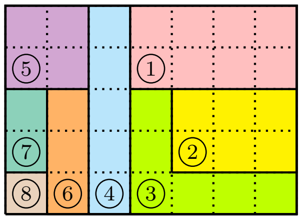



**Figure 2:** Strategy for the $5 \times 7$ field.

## 2 Partially ordered sets

In this section we give a brief primer on partially ordered sets and provide some preliminary definitions. See [**8**, **11**] for a complete treatment of the subject. We introduce the $\mathsf{q}_k$-function which is the central topic of this paper, and explain how it relates to Problem 1.1.

### 2.1 Posets

A *partially ordered set* (or *poset*) is a set $\lambda$ together with a binary relation $\succeq$, which satisfies the following conditions for all $u, v, w \in \lambda$:

(i) $u \succeq u$ (*reflexivity*);

(ii) $u \succeq v$ and $v \succeq u$ imply $u = v$ (*antisymmetricity*);

(iii) $u \succeq v$ and $v \succeq w$ imply $u \succeq w$ (*transitivity*).

We use $a \succ b$ to indicate $a \succeq b$ and $a \neq b$. The order $\succeq$ is a *total order* if either $u \succeq v$ or $v \succeq u$ for all $u, v \in \lambda$. An *order-preserving* map of posets $\lambda, \nu$ is a set map $f : \lambda \to \nu$ such that $f(u) \succeq f(v)$ whenever $u \succeq v$. We say two posets $\lambda, \nu$ are *isomorphic* and write $\lambda \cong \nu$ if there exist mutually inverse order-preserving maps $\lambda \rightleftarrows \nu$.

If $\kappa, \nu$ are posets, then $\kappa \times \nu$ is a poset under the *product partial order*:

$$(x_1, y_1) \succeq (x_2, y_2) \iff x_1 \succeq_\kappa x_2 \text{ and } y_1 \succeq_\nu y_2,$$

for all $x_1, x_2 \in \kappa$ and $y_1, y_2 \in \nu$. Our main examples of posets in this paper are the following:

**Example 2.1.** The *trivial partial order* on a set $\lambda$ has $u \succeq v$ if and only if $u = v$ for all $u, v \in \lambda$. Such a poset is also called an *antichain.*

**Example 2.2.** The natural numbers $\mathbb{N} = \{1, 2, \ldots\}$ are totally ordered under the usual $\geq$ relation, as is any interval $[a, b] = \{a, a+1, \ldots, b\} \subset \mathbb{N}$. In fact, if $\lambda$ is any finite totally ordered set of cardinality $m$, then $\lambda \cong [1, m]$.

**Example 2.3.** Let $m, n \in \mathbb{N}$. Then $[1, m], [1, n]$ are totally ordered sets as in Example 2.2. We write $[\![m, n]\!]$ as shorthand for the poset $[1, m] \times [1, n]$ under the product partial order. If $\kappa, \nu$ are totally ordered sets of cardinality $m, n$ respectively, then $\kappa \times \nu \cong [\![m, n]\!]$.

We represent elements of $[\![m, n]\!]$ as boxes situated in the first quadrant of the plane, arranged so that $(a, b)$ is a box in the $a$th row from the bottom, and in the $b$th column from the left. In this scheme, we have $u \succeq v$ for $u, v \in [\![m, n]\!]$ if and only if the $v$ box is weakly below and to the left (i.e. 'southwest') of the $u$ box. For example, in Figure 3 we show the poset $[\![5, 7]\!]$, with the elements $x = (4, 3), y = (2, 5), z = (2, 2)$. Then we have $x \succeq z, y \succeq z$, with $x, y$ incomparable.

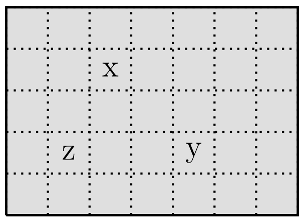



**Figure 3:** The poset $[\![5, 7]\!]$, with elements $x, y, z$

## 2.2 Lower sets and ideals

Let $U$ be a subset of a poset $\lambda$. Then $U$ is itself a poset under the partial order inherited from $\lambda$, and we always assume we take this partial order on $U$. We say $U$ is a *lower set* in $\lambda$ provided that for all $u \in U$, $v \in \lambda$, $u \succeq v$ implies $v \in U$. We say $U$ is a *directed set* in $\lambda$ provided that for all $u, v \in U$, there exists $w \in U$ such that $w \succeq u, v$. We say $U$ is an *ideal* in $\lambda$ if it is a lower set and a directed set. In particular, we allow ideals to be empty.

Let $S, U \subseteq \lambda$. We define subsets:

$$\begin{aligned} S_{\succeq U} &= \{v \in S \mid v \succeq u \text{ for some } u \in U\} \\ S_{\succ U} &= \{v \in S \mid v \succ u \text{ for some } u \in U\} \\ S_{\preceq U} &= \{v \in S \mid v \preceq u \text{ for some } u \in U\} \\ S_{\not\succeq U} &= \{v \in S \mid v \not\succeq u \text{ for all } u \in U\} = S \backslash S_{\succeq U}. \end{aligned}$$

When $U = \{u\}$, we will write $S_{\succeq u}$ in place of $S_{\succeq \{u\}}$, and so on. For any ordered sequence $\boldsymbol{u} = (u_1, \ldots, u_r)$ of elements of $\lambda$, we will also write $S_{\succeq \boldsymbol{u}}$ in place of $S_{\succeq \{u_1, \ldots, u_r\}}$, and so on. We will often apply these definitions with $S = \lambda$.

We will focus primarily on finite posets $\lambda$. In this setting every ideal is either empty or *principal*; i.e. of the form $\lambda_{\preceq u}$ for some $u \in \lambda$, and every lower set is equal to $\lambda_{\preceq U}$ for some $U \subseteq \lambda$.

### 2.3 The $\mathsf{q}_k$-function and Problem 1.1

In this section we define the $\mathsf{q}_k$-function and show that it provides a purely combinatorial rephrasing of Problem 1.1.

**Definition 2.1.** *Let $k \in \mathbb{N}$, and let $\lambda$ be a finite poset. We define the value $\mathsf{q}_k(\lambda) \in \mathbb{Z}_{\geq 0}$ recursively by setting:*

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

*where we implicitly take the partial orders on $\lambda_{\not\succeq u}$, $\lambda_{\succ u}$ to be those inherited from $\lambda$.*

**Example 2.4.** It is easy to check from Definition 2.1 that $\mathsf{q}_k(\lambda) = |\lambda|$ when $|\lambda| \leq 2$. Let $A = \{a, b, c\}$, and consider the posets $\lambda_0, \lambda_1, \lambda_2, \lambda_3$ with underlying set $A$, and partial orders given in terms of Hasse diagrams in Figure 4. That is, we have $y \succeq x$ if

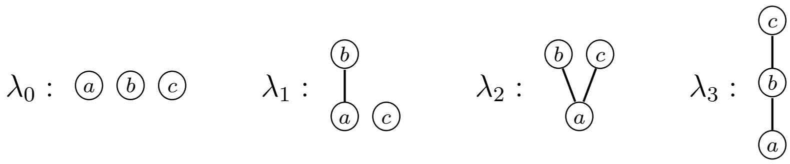



**Figure 4:** Hasse diagrams for $\lambda_0, \lambda_1, \lambda_2, \lambda_3$.

and only if there is an upward path from $x$ to $y$ in the diagram in Figure 4. Note that $\lambda_0$ is the trivial poset on $A$ and $\lambda_3$ is a totally ordered set on $A$. We have $\mathsf{q}_1(\lambda_i) = 3$ for $i = 0, 1, 2, 3$ by definition, and it is straightforward to compute:

$$\mathsf{q}_k(\lambda_0) = 3 \qquad \mathsf{q}_k(\lambda_1) = 2 \qquad \mathsf{q}_k(\lambda_2) = 3 \qquad \mathsf{q}_k(\lambda_3) = 2$$

for all $k \geq 2$.

We now explain how the $\mathsf{q}_k$-function provides a combinatorial rephrasing of Problem 1.1. Recall that in Problem 1.1, $\mu$ is an unknown ideal in $\lambda$ we wish to identify, and we may sequentially query elements of $\lambda$ for membership in $\mu$, with the restriction that we must stop after the $k$th positive query. Note that since $\mu$ is an ideal in a finite set, we have that $\mu = \varnothing$ or $\mu = \lambda_{\preceq x}$ for some $x \in \lambda$. Let $\mathsf{q}'_k(\lambda)$ represent the minimum total number of queries needed to guarantee identification of $\mu$.

**Proposition 2.2.** *We have $\mathsf{q}'_k(\lambda) = \mathsf{q}_k(\lambda)$.*

*Proof.* We first consider the $\lambda = \varnothing$ case. In this case we must have $\mu = \varnothing$, so no queries are needed to identify $\mu$. Thus $\mathsf{q}'_k(\varnothing) = 0 = \mathsf{q}_k(\varnothing)$.

We next consider the $k = 1$ case. With only one positive search query available, the search strategy is very limited. Assume that $u \in \lambda$ and we know $v \notin \mu$ for all $v \succ u$ by previous queries. Then a positive query at $u$ will identify $\mu$ to be the ideal $\lambda_{\preceq u}$. On the other hand, if there exists a element $v \succ u$ whose membership in $\mu$ is unknown, a positive query result at $u$ would result in failure, as $\mu$ could potentially be $\lambda_{\preceq v}$ or $\lambda_{\preceq u}$, and we would be left with no further queries to distinguish these possibilities. We see then that the only permissible search strategy is to query all of the elements of $\lambda$ in some non-increasing sequence, where the first positive query result will identify the generator of the ideal $\mu$. If $\mu = \varnothing$, the ideal will only be identified after the final (negative) query, so we have $\mathsf{q}'_k(\lambda) = |\lambda| = \mathsf{q}_k(\lambda)$.

Finally, we consider the general $k > 1$, $|\lambda| > 0$ case. By induction, assume that $\mathsf{q}'_\ell(\nu) = \mathsf{q}_\ell(\nu)$ for all $\ell < k$ or $|\nu| < |\lambda|$. Assume the first query is at some element $u \in \lambda$. If the query is negative, this implies that $\mu \subseteq \lambda_{\not\succeq u}$, and we still have $k$ positive queries to work with. By induction, the minimal total number of queries necessary to guarantee identification of $\mu$ in $\lambda_{\not\succeq u}$ is $\mathsf{q}_k(\lambda_{\not\succeq u})$.

On the other hand, assume the query at $u \in \lambda$ is positive. This implies that the ideal generator $x$ could be any element in $\lambda_{\succeq u}$, and we now have $k-1$ positive query results remaining. Let $\mu'$ be the ideal $\mu \cap \lambda_{\succ u}$ in $\lambda_{\succ u}$. Then we have $\mu' = \varnothing$ if and only if $x = u$, and $\mu'$ is nonempty if and only if $x \in \lambda_{\succ u}$ and $\mu' = (\lambda_{\succ u})_{\preceq x}$. Therefore, identifying $\mu$ is equivalent to identifying the ideal $\mu'$ in $\lambda_{\succ u}$. By induction, $\mathsf{q}_{k-1}(\lambda_{\succ u})$ is the minimal total number of queries necessary to guarantee success in this search.

Therefore if we begin by querying $u$, the minimal number of queries that will be necessary to guarantee identification of $\mu$ in $\lambda$ is $\mathsf{q}_k(\lambda_{\not\succeq u})+1$ if $u \notin \mu$, and $\mathsf{q}_{k-1}(\lambda_{\succ u})+1$ if $u \in \mu$. Thus, by first querying $u$, the minimal number of queries necessary is

$$\max\{\mathsf{q}_k(\lambda_{\not\succeq u}), \mathsf{q}_{k-1}(\lambda_{\succ u})\} + 1.$$

Therefore, taking the minimum over all possible choices of the initial query $u$, we have that $\mathsf{q}'_k(\lambda) = \mathsf{q}_k(\lambda)$, as desired. □

## 3 Binomial sums and triangular numbers

Bounds for the $\mathsf{q}_k$-function will be shown to be directly related to binomial sums, and, in the $k = 2$ case, triangular numbers. In preparation for establishing this fact, we investigate some properties of binomial sums, and triangular numbers in particular.

### 3.1 Binomial sums

Throughout this section, we fix $k \in \mathbb{N}$.

**Definition 3.1.** *Define the function* $T_k : \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ *via:*

$$T_k(x) = \sum_{i=1}^{k} \binom{x}{i}.$$

Notably, when $k = 1$ we have $T_1(x) = x$, and when $k = 2$ we have

$$T_2(x) = \frac{x(x+1)}{2} = 1 + 2 + \cdots + x, \tag{3.1}$$

the $x$th *triangular number*. The following lemma is clear.

**Lemma 3.2.** *For any* $k \in \mathbb{N}$, $x, y \in \mathbb{Z}_{\geq 0}$, *we have* $x < y$ *if and only if* $T_k(x) < T_k(y)$.

The next function is key in describing lower bounds for the $\mathsf{q}_k$-function.

**Definition 3.3.** *Define the function* $\tau_k : \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ *by setting* $\tau_k(x)$ *to be the unique non-negative integer such that*

$$T_k(\tau_k(x) - 1) < x \leq T_k(\tau_k(x)).$$

The next two lemmas are clear from definitions.

**Lemma 3.4.** *For any* $x \in \mathbb{Z}_{\geq 0}$, *we have* $\tau_k(T_k(x)) = x$.

**Lemma 3.5.** *For any* $k \in \mathbb{N}$, $x \leq y$, *we have* $\tau_k(x) \leq \tau_k(y)$.

We now prove some additional useful technical lemmas on $T_k$ and $\tau_k$.

**Lemma 3.6.** *For all* $x > 0$, *we have* $T_k(x) = T_k(x-1) + T_{k-1}(x-1) + 1$.

*Proof.* We have

$$\begin{aligned} 1 + T_k(x-1) + T_{k-1}(x-1) &= \binom{x-1}{0} + \sum_{i=1}^{k} \binom{x-1}{i} + \sum_{i=1}^{k-1} \binom{x-1}{i} \\ &= \sum_{i=1}^{k} \left[ \binom{x-1}{i} + \binom{x-1}{i-1} \right] = \sum_{i=1}^{k} \binom{x}{i} = T_k(x), \end{aligned}$$

where the third equality follows from the binomial recurrence relation. ☐

**Lemma 3.7.** *Let* $x, y \in \mathbb{Z}_{\geq 0}$ *be such that* $y < T_{k-1}(\tau_k(x) - 2) + 2$. *Then* $\tau_k(x) - 1 \leq \tau_k(x - y)$.

*Proof.* We have by Lemma 3.6 that

$$T_k(\tau_k(x) - 2) = T_k(\tau_k(x) - 1) - T_{k-1}(\tau_k(x) - 2) - 1 < T_k(\tau_k(x) - 1) - y + 1.$$

By the definition of $\tau_k(x)$ we have $T_k(\tau_k(x) - 1) < x$, so

$$T_k(\tau_k(x) - 2) \leq T_k(\tau_k(x) - 1) - y < x - y.$$

Then by the definition of $\tau_k(x - y)$, we have $\tau_k(x - y) > \tau_k(x) - 2$. Thus $\tau_k(x - y) \geq \tau_k(x) - 1$. ☐

**Lemma 3.8.** *If* $x, y \in \mathbb{Z}_{\geq 0}$ *and* $n \in \mathbb{N}$ *are such that* $x \equiv 0 \pmod{n}$ *and* $T_k(\tau_k(x)) \equiv y \pmod{n}$, *where* $0 \leq y < n$, *then* $T_k(\tau_k(x)) - x \geq y$.

*Proof.* By definition, $T_k(\tau_k(x)) \geq x$. Then we have $T_k(\tau_k(x)) - x \equiv y \pmod{n}$, and $T_k(\tau_k(x)) - x \geq 0$, so $T_k(\tau_k(x)) - x = y + nt$ for some $t \in \mathbb{Z}_{\geq 0}$, so the result follows. □

## 3.2 Triangular numbers

Now we prove some technical lemmas in the case $k = 2$, recalling that $T_2(x)$ is the triangular number $1 + \cdots + x$. The next lemma is just a special case of Lemma 3.7.

**Lemma 3.9.** *If* $x, y \in \mathbb{Z}_{\geq 0}$, *with* $y < \tau_2(x)$, *then* $\tau_2(x) - 1 \leq \tau_2(x - y)$.

**Lemma 3.10.** *Let* $x, y \in \mathbb{Z}_{\geq 0}, \ell \in \mathbb{N}$, *with* $0 \leq y \leq x - \ell\tau_2(x) + T_2(\ell - 1)$. *Then we have*

$$\tau_2(y) \leq \tau_2(x) - \ell.$$

*Proof.* By Definition 3.3, we have

$$\begin{aligned} y &\leq x - \ell\tau_2(x) + T_2(\ell - 1) \leq T_2(\tau_2(x)) - \ell\tau_2(x) + T_2(\ell - 1) \\ &= [1 + \cdots + \tau_2(x)] - \ell\tau_2(x) + [1 + 2 + \cdots + (\ell - 1)] \\ &= [1 + 2 + \cdots + \tau_2(x)] - [(\tau_2(x) - (\ell - 1)) + \cdots + (\tau_2(x) - 1) + \tau_2(x)] \\ &= 1 + 2 + \cdots + (\tau_2(x) - \ell) = T_2(\tau_2(x) - \ell). \end{aligned}$$

Then, applying $\tau_2$ to both sides of the inequality, we have by Lemmas 3.4 and 3.5 that

$$\tau_2(y) \leq \tau_2(T_2(\tau_2(x) - \ell)) = \tau_2(x) - \ell,$$

as desired. □

**Lemma 3.11.** *Let* $y, r, n \in \mathbb{N}$, *with* $y \equiv r \pmod{n}$. *Then:*

$$T_2(y) \equiv \begin{cases} T_2(r) + \frac{n}{2} \pmod{n} & \text{if } n \equiv 0 \pmod{2}, \frac{y-r}{n} \equiv 1 \pmod{2}; \\ T_2(r) \pmod{n} & \text{otherwise.} \end{cases}$$

*Proof.* We may assume without loss of generality that $y \geq r$. Note that since $y \equiv r \pmod{n}$, we have $y - r = n\ell$ for some $\ell \in \mathbb{Z}_{\geq 0}$. We prove the claim by induction on $\ell$. Let $\ell = 0$. Then $y = r$ and $\frac{y-r}{n} \equiv 0 \pmod{2}$. Therefore, $T_2(y) = T_2(r) \equiv T_2(r) \pmod{n}$ so the base case holds.

Now assume $\ell > 0$ and the claim holds for all $\ell' < \ell$. Then

$$\begin{aligned}
T_2(y) &= T_2(r+n\ell) \\
&= (r+n\ell) + (r+n\ell-1) + \cdots + (r+n\ell-(n-1)) + T_2(r+n(\ell-1)) \\
&= nr + n^2\ell - (0+\cdots+(n-1)) + T_2(r+n(\ell-1)) \\
&= nr + n^2\ell - T_2(n-1) + T_2(r+n(\ell-1)) \\
&= nr + n^2\ell - \frac{(n-1)n}{2} + T_2(r+n(\ell-1)) \\
&\equiv -\frac{(n-1)n}{2} + T_2(r+n(\ell-1)) \pmod{n}.
\end{aligned}$$

We consider three separate cases, based on the parity of $n$ and $\ell$.

*Case 1.* Suppose $n$ is odd. Then we have that $T_2(r+n(\ell-1)) \equiv T_2(r) \pmod{n}$ by the induction assumption. Therefore,

$$T_2(y) \equiv -\frac{(n-1)n}{2} + T_2(r+n(\ell-1)) \equiv -n\cdot\frac{(n-1)}{2} + T_2(r) \equiv T_2(r) \pmod{n}.$$

*Case 2.* Suppose $n$ is even and $\ell$ is odd. Then $\ell-1$ is even, so then we have $T_2(r+n(\ell-1)) \equiv T_2(r) \pmod{n}$ by the induction assumption. Then

$$\begin{aligned}
T_2(y) &\equiv -\frac{(n-1)n}{2} + T_2(r+n(\ell-1)) \equiv -\frac{n}{2}(n-1) + T_2(r) \pmod{n} \\
&\equiv -\frac{n}{2}(-1) + T_2(r) \equiv T_2(r) + \frac{n}{2} \pmod{n}.
\end{aligned}$$

*Case 3.* Suppose $n$ is even and $\ell$ is even. Then $\ell-1$ is odd, so then we have $T_2(r+n(\ell-1)) \equiv T_2(r) + \frac{n}{2} \pmod{n}$ by the induction assumption. Then

$$\begin{aligned}
T_2(y) &\equiv -\frac{(n-1)n}{2} + T_2(r+n(\ell-1)) \pmod{n} \\
&\equiv -\frac{n}{2}(n-1) + T_2(r) + \frac{n}{2} \equiv n\left(1-\frac{n}{2}\right) + T_2(r) \equiv T_2(r) \pmod{n}.
\end{aligned}$$

Thus in any case, the claim holds for $\ell$, completing the induction step and the proof. □

## 4 Bounds on the $\mathsf{q}_k$-function

Now we establish bounds on the $\mathsf{q}_k$-function. The following lemma is clear from Definition 2.1.

**Lemma 4.1.** *If $\lambda \cong \nu$, then $\mathsf{q}_k(\lambda) = \mathsf{q}_k(\nu)$.*

**Theorem 4.2.** *For all $\lambda, k$, we have $\tau_k(|\lambda|) \leq \mathsf{q}_k(\lambda) \leq |\lambda|$.*

*Proof.* We first prove that $\mathsf{q}_k(\lambda) \le |\lambda|$. The claim holds for $k = 1$ and $\lambda = \varnothing$ by Definition 2.1. Now let $k > 1$, $|\lambda| > 0$, and assume $\mathsf{q}_{k'}(\lambda') \le |\lambda'|$ for all $k' < k$, $|\lambda'| < |\lambda|$. Let $v$ be any maximal element in $\lambda$. Then we have $\lambda_{\succ v} = \varnothing$ and $|\lambda_{\not\succeq v}| = |\lambda| - 1$, so:

$$\begin{aligned}\mathsf{q}_k(\lambda) &= \min\{\max\{\mathsf{q}_k(\lambda_{\not\succeq u}), \mathsf{q}_{k-1}(\lambda_{\succ u})\} \mid u \in \lambda\} + 1 \\ &\le \max\{\mathsf{q}_k(\lambda_{\not\succeq v}), \mathsf{q}_{k-1}(\lambda_{\succ v})\} + 1 \le \max\{|\lambda_{\not\succeq v}|, 0\} + 1 = (|\lambda| - 1) + 1 = |\lambda|,\end{aligned}$$

as desired.

Now we prove that $\tau_k(|\lambda|) \le \mathsf{q}_k(\lambda)$. The claim holds for $k = 1$, as $\mathsf{q}_1(\lambda) = |\lambda| = \binom{|\lambda|}{1} = T_1(|\lambda|)$, and the claim holds for $\lambda = \varnothing$, as we have $\mathsf{q}_k(\varnothing) = 0 = \sum_{i=1}^{k} \binom{0}{i} = T_k(0)$. Now let $k > 1$, $|\lambda| > 0$, and assume $\tau_{k'}(|\lambda'|) \le \mathsf{q}_{k'}(\lambda')$ for all $k' < k$ or $|\lambda'| < |\lambda|$. For some $u \in \lambda$, we have

$$\mathsf{q}_k(\lambda) = \max\{\mathsf{q}_k(\lambda_{\not\succeq u}), \mathsf{q}_{k-1}(\lambda_{\succ u})\} + 1.$$

Then by the induction assumption we have

$$\mathsf{q}_k(\lambda) \ge \mathsf{q}_k(\lambda_{\not\succeq u}) + 1 \ge \tau_k(|\lambda_{\not\succeq u}|) + 1 \tag{4.1}$$

and

$$\mathsf{q}_k(\lambda) \ge \mathsf{q}_{k-1}(\lambda_{\succ u}) + 1 \ge \tau_{k-1}(|\lambda_{\succ u}|) + 1. \tag{4.2}$$

Assume by way of contradiction that $\mathsf{q}_k(\lambda) < \tau_k(|\lambda|)$. First we claim that $|\lambda_{\succ u}| < T_{k-1}(\tau_k(|\lambda|) - 2) + 1$. Indeed, if $|\lambda_{\succ u}| \ge T_{k-1}(\tau_k(|\lambda|) - 2) + 1$, then by Definition 3.3 we would have

$$T_{k-1}(\tau_k(|\lambda|) - 2) < T_{k-1}(\tau_k(|\lambda|) - 2) + 1 \le |\lambda_{\succ u}| \le T_{k-1}(\tau_{k-1}(|\lambda_{\succ u}|)).$$

Then Lemma 3.2 implies that $\tau_{k-1}(|\lambda_{\succ u}|) > \tau_k(|\lambda|) - 2$, so $\tau_{k-1}(|\lambda_{\succ u}|) \ge \tau_k(|\lambda|) - 1$. But then

$$\tau_{k-1}(|\lambda_{\succ u}|) + 1 \ge \tau_k(|\lambda|) > \mathsf{q}_k(\lambda),$$

a contradiction of (4.2). Thus $|\lambda_{\succ u}| < T_{k-1}(\tau_k(|\lambda|) - 2) + 1$ as desired.

Note then that $|\lambda_{\succ u}| + 1 < T_{k-1}(\tau_k(|\lambda|) - 2) + 2$, so by Lemma 3.7, we have

$$\tau_k(|\lambda|) - 1 \le \tau_k(|\lambda| - |\lambda_{\succ u}| - 1).$$

Therefore, applying (4.1) we have

$$\mathsf{q}_k(\lambda) \ge \tau_k(|\lambda_{\not\succeq u}|) + 1 = \tau_k(|\lambda| - |\lambda_{\succ u}| - 1) + 1 \ge (\tau_k(|\lambda|) - 1) + 1 = \tau_k(|\lambda|) > \mathsf{q}_k(\lambda),$$

a contradiction. Therefore $\tau_k(|\lambda|) \le \mathsf{q}_k(\lambda)$, as desired. This completes the induction step, and the proof. □

With the following two lemmas, we prove that the bounds of Theorem 4.2 are tight with respect to arbitrary posets.

**Lemma 4.3.** *Let $\lambda$ be a poset with trivial partial order. Then $\mathsf{q}_k(\lambda) = |\lambda|$.*

*Proof.* If $\lambda = \varnothing$ or $k = 1$, the claim follows by Definition 2.1. Now let $k > 1$, $|\lambda| > 0$, and assume $\mathsf{q}_{k'}(\lambda') = |\lambda'|$ for all $k' < k$, and trivial posets $\lambda'$ with $|\lambda'| < |\lambda|$. Let $u \in \lambda$. Then we have that $\lambda_{\succ u} = \varnothing$, and $\lambda_{\not\succeq u} = \lambda\backslash\{u\}$ is itself a trivial poset. Therefore by the induction assumption we have

$$\begin{aligned}\mathsf{q}_k(\lambda) &= \min\{\max\{\mathsf{q}_k(\lambda_{\not\succeq u}), \mathsf{q}_{k-1}(\lambda_{\succ u})\} \mid u \in \lambda\} + 1 \\ &= \min\{\max\{|\lambda| - 1, 0\} \mid u \in \lambda\} + 1 = (|\lambda| - 1) + 1 = |\lambda|,\end{aligned}$$

as desired. □

**Lemma 4.4.** *Let $\lambda$ be a totally ordered set. Then $\mathsf{q}_k(\lambda) = \tau_k(|\lambda|)$.*

*Proof.* As usual, we note that the claim holds for $k = 1, \lambda = \varnothing$ by Definition 2.1. We now let $k > 1$ and $|\lambda| > 0$, and make the induction assumption that $\mathsf{q}_{k'}(\lambda') = \tau_{k'}(|\lambda'|)$ for all $k' < k$ and totally ordered $\lambda'$ with $|\lambda'| < |\lambda|$.

We may assume $\lambda = [1, n]$, as any totally ordered set of cardinality $n$ is equivalent to this interval. Note that we have $0 \leq T_k(\tau_k(n) - 1) < n$ by Definition 3.3, so $v := T_k(\tau_k(n) - 1) + 1 \in [1, n]$. Then, applying Lemma 3.4, we have

$$\mathsf{q}_k(\lambda_{\not\succeq v}) = \tau_k(|[1, v - 1]|) = \tau_k(v - 1) = \tau_k(T_k(\tau_k(n) - 1)) = \tau_k(n) - 1.$$

On the other hand, we have

$$\mathsf{q}_{k-1}(\lambda_{\succ v}) = \tau_{k-1}(|[v + 1, n]|) = \tau_{k-1}(n - v) = \tau_{k-1}(n - T_k(\tau_k(n) - 1) - 1)).$$

Then we have

$$\begin{aligned}\mathsf{q}_{k-1}(\lambda_{\succ v}) &= \tau_{k-1}(n - T_k(\tau_k(n) - 1) - 1)) \leq \tau_{k-1}(T_k(\tau_k(n)) - T_k(\tau_k(n) - 1) - 1)) \\ &= \tau_{k-1}((T_{k-1}(\tau_k(n) - 1) + 1) - 1) = \tau_{k-1}(T_{k-1}(\tau_k(n) - 1)) = \tau_k(n) - 1,\end{aligned}$$

using Lemma 3.5 and the fact that $n \leq T_k(\tau_k(n))$ by Definition 3.3 for the first inequality, Lemma 3.6 for the second equality, and Lemma 3.4 for the last equality.

Thus we have

$$\begin{aligned}\mathsf{q}_k(\lambda) &= \min\{\max\{\mathsf{q}_k(\lambda_{\not\succeq u}), \mathsf{q}_{k-1}(\lambda_{\succ u})\} + 1 \mid u \in \lambda\} \\ &\leq \max\{\mathsf{q}_k(\lambda_{\not\succeq v}), \mathsf{q}_{k-1}(\lambda_{\succ v})\} + 1 \leq (\tau_k(n) - 1) + 1 = \tau_k(n) = \tau_k(|\lambda|).\end{aligned}$$

Since $\mathsf{q}_k(\lambda) \geq \tau_k(|\lambda|)$ by Theorem 4.2, we have $\mathsf{q}_k(\lambda) = \tau_k(|\lambda|)$. This completes the induction step, and the proof. □

**Remark 4.3.** The proof of Lemma 4.4 contains a solution to the strategy question from Problem 1.1 for totally ordered sets, defined recursively for any $k \in \mathbb{N}$. Namely, one should query the element $v$ such that $|\lambda_{\prec v}| = T_k(\tau_k(|\lambda|) - 1)$. If the query is negative, repeat the process with the totally ordered set $\lambda_{\prec v}$. If the query is positive and $k = 1$, stop. Otherwise, repeat the process with the totally ordered set $\lambda_{\succ v}$ and $k := k - 1$. The final positive query will identify the element which generates the ideal $\mu$.

**Remark 4.4.** In view of Theorem 4.2 and Lemmas 4.3 and 4.4, one may be led to conjecture that $\mathsf{q}_k(\lambda') \leq \mathsf{q}_k(\lambda)$ when $\lambda'$ is a refinement of the poset $\lambda$. This does not hold in general, however. For a counterexample, see Example 2.4, where the posets $\lambda_0, \lambda_1, \lambda_2, \lambda_3$ are sequential refinements, but the corresponding sequence of $\mathsf{q}_k$ values is not monotonic when $k \geq 2$.

# 5 Strategy in the $k = 2$ case

We will now narrow our focus to the $k = 2$ setting. We develop a combinatorial language for describing query strategies in response to Problem 1.1. We fix some nonempty finite poset $\lambda$ throughout this section.

**Definition 5.1.** *Let $r \in \mathbb{N}$, and $\boldsymbol{u} = (u_1, \ldots, u_r)$ be a sequence of elements of $\lambda$. For each $t = 1, \ldots, r$, define the subset:*

$$\lambda_{\boldsymbol{u}}^{(t)} := \lambda_{\succeq u_t} \backslash \lambda_{\succeq \{u_1, \ldots, u_{t-1}\}} = \{v \in \lambda \mid v \succeq u_t, v \not\succeq u_i \text{ for all } i = 1, \ldots, t-1\}.$$

*If $\lambda_{\succeq \boldsymbol{u}} = \lambda$ and $\lambda_{\boldsymbol{u}}^{(t)} \neq \varnothing$ for all $t = 1, \ldots, r$, we call $\boldsymbol{u}$ a* $\lambda$-strategy.

By definition, the sets $\lambda_{\boldsymbol{u}}^{(1)}, \ldots, \lambda_{\boldsymbol{u}}^{(r)}$ are mutually disjoint, so if $\boldsymbol{u}$ is a $\lambda$-strategy, we have:

$$\lambda = \lambda_{\boldsymbol{u}}^{(1)} \sqcup \cdots \sqcup \lambda_{\boldsymbol{u}}^{(r)}. \tag{5.1}$$

## 5.1 The $\mathbf{Q}_2$-function

**Definition 5.2.** *For a sequence of elements $\boldsymbol{u} = (u_1, \ldots, u_r)$ in $\lambda$, we define:*

$$\mathsf{Q}_2(\lambda, \boldsymbol{u}) := \max\{|\lambda_{\boldsymbol{u}}^{(t)}| + t - 1 \mid t = 1, \ldots, r\}.$$

We will primarily be concerned with the value of $\mathsf{Q}_2(\lambda, \boldsymbol{u})$ when $\boldsymbol{u}$ is a $\lambda$-strategy.

**Example 5.2.** Let $\lambda = [\![5, 7]\!]$, and define the $\lambda$-strategy

$$\boldsymbol{u} = ((2,6), (5,2), (1,5), (3,3), (2,1), (1,4), (1,1)).$$

Then we may visually represent $\boldsymbol{u}$ in as in Figure 5. The elements $u_1, \ldots, u_7$ are marked with circled numbers. For each $i \in \{1, \ldots, 7\}$, $\lambda_{\boldsymbol{u}}^{(i)}$ is the set of boxes in the same colored region as the box marked $i$. The cardinalities of these sets are $8, 4, 6, 4, 9, 1, 3$ respectively, so we have

$$\mathsf{Q}_2(\lambda, \boldsymbol{u}) = \max\{8+0, 4+1, 6+2, 4+3, 9+4, 1+5, 3+6\} = 13.$$

We consider now some special choices of $\lambda$-strategies.

**Lemma 5.3.** *For any nonempty finite poset $\lambda$, let $\boldsymbol{u} = (u_1, \ldots, u_{|\lambda|})$ be a linear extension of the partial order on $\lambda$, i.e., an arrangement of the elements of $\lambda$ such that $i < j$ whenever $u_i \succ u_j$. Then $\boldsymbol{u}$ is a $\lambda$-strategy and $\mathsf{Q}_2(\lambda, \boldsymbol{u}) = |\lambda|$.*

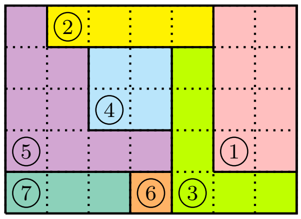



**Figure 5:** The poset $\lambda = [\![5, 7]\!]$ with $\lambda$-strategy $\boldsymbol{u}$.

*Proof.* By the condition on $\boldsymbol{u}$ we have $|\lambda_{\boldsymbol{u}}^{(t)}| = 1$ for all $t$, so $\boldsymbol{u}$ is a $\lambda$-strategy and

$$\begin{aligned}\mathsf{Q}_2(\lambda, \boldsymbol{u}) &= \max\{|\lambda_{\boldsymbol{u}}^{(t)}| + t - 1 \mid t \\ &= 1, \ldots, |\lambda|\} = \max\{1 + t - 1 \mid t = 1, \ldots, |\lambda|\} = |\lambda|,\end{aligned}$$

as desired. □

**Lemma 5.4.** *Let $\lambda$ be a nonempty finite poset, and assume there exists a $\lambda$-strategy $\boldsymbol{u} = (u_1)$ of length one. Then we have $\mathsf{Q}_2(\lambda, \boldsymbol{u}) = |\lambda|$.*

*Proof.* By the definition of $\lambda$-strategies $\boldsymbol{u}$, we must have $\lambda = \lambda_{\succeq \boldsymbol{u}} = \lambda_{\succeq u_1} = \lambda_{\boldsymbol{u}}^{(1)}$. Thus we have $\mathsf{Q}_2(\lambda, \boldsymbol{u}) = |\lambda_{\boldsymbol{u}}^{(1)}| = |\lambda|$, as desired. □

**Remark 5.3.** Note that $\{\mathsf{Q}_2(\lambda, \boldsymbol{u}) \mid \boldsymbol{u} \text{ a } \lambda\text{-strategy}\}$ is by definition a nonempty subset of $\mathbb{Z}_{\geq 0}$, and thus possesses a unique minimum value, which by the above lemmas is less than or equal to $|\lambda|$.

For sequences of elements $\boldsymbol{v} = (v_1, \ldots, v_s)$ and $\boldsymbol{w} = (w_1, \ldots, w_r)$ in $\lambda$, we will write $\boldsymbol{vw}$ for the concatenation $(v_1, \ldots, v_s, w_1, \ldots, w_r)$, or just $v_1\boldsymbol{w}$ if $\boldsymbol{v} = (v_1)$. For $u \in \lambda$ with $\lambda_{\succeq u} \neq \lambda$, note that $u\boldsymbol{w}$ is a $\lambda$-strategy if and only if $\boldsymbol{w}$ is a $\lambda_{\not\succeq u}$-strategy.

**Lemma 5.5.** *Let $\lambda$ be a nonempty poset. Let $\boldsymbol{v} = (v_1, \ldots, v_s)$ be a sequence of elements of $\lambda$, and $\boldsymbol{w} = (w_1, \ldots, w_r)$ be a sequence of elements of $\lambda_{\not\succeq \boldsymbol{v}}$. Then, setting $\boldsymbol{u} = \boldsymbol{vw}$, we have*

$$\mathsf{Q}_2(\lambda, \boldsymbol{u}) = \max\{\mathsf{Q}_2(\lambda, \boldsymbol{v}), \mathsf{Q}_2(\lambda_{\not\succeq \boldsymbol{v}}, \boldsymbol{w}) + s\}.$$

*Proof.* Note that for $t = 1, \ldots, s$, we have $\lambda_{\boldsymbol{u}}^{(t)} = \lambda_{\boldsymbol{v}}^{(t)}$, and for $t = s+1, \ldots, s+r$, we have

$$\begin{aligned}\lambda_{\boldsymbol{u}}^{(t)} &= \lambda_{\succeq u_t} \backslash \lambda_{\succeq \{u_1, \ldots, u_{t-1}\}} = (\lambda_{\not\succeq \{u_1, \ldots, u_s\}})_{\succeq u_t} \backslash (\lambda_{\not\succeq \{u_1, \ldots, u_s\}})_{\succeq \{u_{s+1}, \ldots, u_{t-1}\}} \\ &= (\lambda_{\not\succeq \{v_1, \ldots, v_s\}})_{\succeq w_{t-s}} \backslash (\lambda_{\succeq \{v_1, \ldots, v_s\}})_{\succeq \{w_1, \ldots, w_{t-s-1}\}} = (\lambda_{\not\succeq \boldsymbol{v}})_{\boldsymbol{w}}^{(t-s)}.\end{aligned}$$

Thus we have

$$\begin{aligned}
\mathsf{Q}_2(\lambda, \boldsymbol{u}) &= \max\{|\lambda_{\boldsymbol{u}}^{(t)}| + t - 1 \mid t = 1, \dots, s + r\} \\
&= \max\{\max\{|\lambda_{\boldsymbol{u}}^{(t)}| + t - 1 \mid t = 1, \dots, s\}, \\
&\qquad\quad \max\{|\lambda_{\boldsymbol{u}}^{(t)}| + t - 1 \mid u = s + 1, \dots, s + r\}\} \\
&= \max\{\max\{|\lambda_{\boldsymbol{v}}^{(t)}| + t - 1 \mid t = 1, \dots, r\}, \\
&\qquad\quad \max\{|(\lambda_{\not\succeq \boldsymbol{v}})_{\boldsymbol{w}}^{(t-s)}| + t - 1 \mid u = s + 1, \dots, s + r\}\} \\
&= \max\{\mathsf{Q}_2(\lambda, \boldsymbol{v}), \max\{|(\lambda_{\not\succeq \boldsymbol{v}})_{\boldsymbol{w}}^{(t)}| + t + s - 1 \mid t = 1, \dots, r\}\} \\
&= \max\{\mathsf{Q}_2(\lambda, \boldsymbol{v}), \max\{|(\lambda_{\not\succeq \boldsymbol{v}})_{\boldsymbol{w}}^{(t)}| + t - 1 \mid t = 1, \dots, r\} + s\} \\
&= \max\{\mathsf{Q}_2(\lambda, \boldsymbol{v}), \mathsf{Q}_2(\lambda_{\not\succeq \boldsymbol{v}}, \boldsymbol{w}) + s\},
\end{aligned}$$

as desired. □

## 5.2 Connecting $\mathbf{Q}_2$ and $\mathbf{q}_2$

**Theorem 5.6.** *Let $\lambda$ be a nonempty poset. We have*

$$\mathsf{q}_2(\lambda) = \min\{\mathsf{Q}_2(\lambda, \boldsymbol{u}) \mid \boldsymbol{u} \text{ a } \lambda\text{-strategy}\}. \tag{5.4}$$

*Proof.* We go by induction on $|\lambda|$. The base case $|\lambda| = 1$ follows immediately from Lemma 5.4. Now assume $|\lambda| > 1$ and the claim holds for all $|\lambda'| < |\lambda|$. Note that by Lemmas 5.3 and 5.4, it suffices to take the minimum on the right of (5.4) over $\lambda$-strategies of length greater than one. Thus we have

$$\begin{aligned}
&\min\{\mathsf{Q}_2(\lambda, \boldsymbol{u}) \mid \boldsymbol{u} \text{ a } \lambda\text{-strategy}\} \\
&\quad= \min\{\mathsf{Q}_2(\lambda, \boldsymbol{u}) \mid \boldsymbol{u} \text{ a } \lambda\text{-strategy of length greater than one}\} \\
&\quad= \min\{\mathsf{Q}_2(\lambda, u\boldsymbol{w}) \mid u \in \lambda, u\boldsymbol{w} \text{ a } \lambda\text{-strategy}\} \\
&\quad= \min\{\mathsf{Q}_2(\lambda, u\boldsymbol{w}) \mid u \in \lambda, \boldsymbol{w} \text{ a } \lambda_{\not\succeq u}\text{-strategy}\} \\
&\quad= \min\{\max\{\mathsf{Q}_2(\lambda, (u)), \mathsf{Q}_2(\lambda_{\not\succeq u}, \boldsymbol{w}) + 1\} \mid u \in \lambda, \boldsymbol{w} \text{ a } \lambda_{\not\succeq u}\text{-strategy}\} \\
&\quad= \min\{\max\{|\lambda_{\succeq u}|, \mathsf{Q}_2(\lambda_{\not\succeq u}, \boldsymbol{w}) + 1\} \mid u \in \lambda, \boldsymbol{w} \text{ a } \lambda_{\not\succeq u}\text{-strategy}\} \\
&\quad= \min\{\max\{|\lambda_{\succ u}| + 1, \mathsf{Q}_2(\lambda_{\not\succeq u}, \boldsymbol{w}) + 1\} \mid u \in \lambda, \boldsymbol{w} \text{ a } \lambda_{\not\succeq u}\text{-strategy}\} \\
&\quad= \min\{\max\{\mathsf{q}_1(\lambda_{\succ u}) + 1, \mathsf{Q}_2(\lambda_{\not\succeq u}, \boldsymbol{w}) + 1\} \mid u \in \lambda, \boldsymbol{w} \text{ a } \lambda_{\not\succeq u}\text{-strategy}\} \\
&\quad= \min\{\min\{\max\{\mathsf{q}_1(\lambda_{\succ u}) + 1, \mathsf{Q}_2(\lambda_{\not\succeq u}, \boldsymbol{w}) + 1\} \mid \boldsymbol{w} \text{ a } \lambda_{\not\succeq u}\text{-strategy}\} \mid u \in \lambda\} \\
&\quad= \min\{\max\{\mathsf{q}_1(\lambda_{\succ u}) + 1, \min\{\mathsf{Q}_2(\lambda_{\not\succeq u}, \boldsymbol{w}) \mid \boldsymbol{w} \text{ a } \lambda_{\not\succeq u}\text{-strategy}\} + 1\} \mid u \in \lambda\} \\
&\quad= \min\{\max\{\mathsf{q}_1(\lambda_{\succ u}) + 1, \mathsf{q}_2(\lambda_{\not\succeq u}) + 1\} \mid u \in \lambda\} \\
&\quad= \min\{\max\{\mathsf{q}_2(\lambda_{\not\succeq u}), \mathsf{q}_1(\lambda_{\succ u})\} \mid u \in \lambda\} + 1 \\
&\quad= \mathsf{q}_2(\lambda).
\end{aligned}$$

The fourth equality above follows from Lemma 5.5, and the tenth equality follows from the induction assumption. This completes the induction step, and the proof. □

### 5.3 Some examples

Combining Theorems 4.2 and 5.6 can be a useful method of computing $\mathsf{q}_2(\lambda)$, as shown in the examples below.

**Example 5.5.** Let $\lambda = [\![5,7]\!]$, and consider the $\lambda$-strategy:

$$\boldsymbol{u} = ((4,4),(2,5),(1,4),(1,3),(4,1),(1,2),(2,1),(1,1)).$$

Then, as in Example 5.2, we visually represent $\boldsymbol{u}$ in Figure 6. This gives

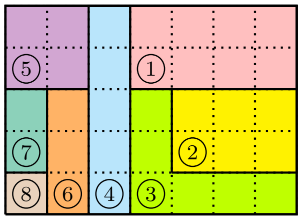



**Figure 6:** The poset $\lambda = [\![5,7]\!]$ with $\lambda$-strategy $\boldsymbol{u}$.

$$\mathsf{Q}_2(\lambda,\boldsymbol{u}) = \max\{8+0,6+1,6+2,5+3,4+4,3+5,2+6,1+7\} = 8.$$

Thus by Theorem 5.6 we have $\mathsf{q}_2(\lambda) \leq 8$. But by Theorem 4.2 we also have

$$\mathsf{q}_2(\lambda) \geq \tau_2(|\lambda|) = \tau_2(35) = 8,$$

so $\mathsf{q}_2(\lambda) = 8$.

**Example 5.6.** Let $\lambda = [\![6,6]\!]$. As $|\lambda| = 36$, any $\lambda$-strategy $\boldsymbol{u} = (u_1,\ldots,u_r)$ which satisfies $\mathsf{Q}_2(\lambda,\boldsymbol{u}) = \tau_2(|\lambda|) = 8$ must have $r = 8$ and $|\lambda_{\boldsymbol{u}}^{(t)}| = 9-t$ for all $t = 1,\ldots,8$. It is straightforward to check that no such $\lambda$-strategy exists, so by Theorems 4.2 and 5.6, we have $\mathsf{q}_2(\lambda,\boldsymbol{u}) > 8$. Now consider the $\lambda$-strategy :

$$\boldsymbol{v} = ((5,3),(4,2),(2,4),(3,1),(1,4),(2,1),(1,2),(1,1)).$$

We visually represent $\boldsymbol{v}$ in Figure 7. This gives $\mathsf{Q}_2(\lambda,\boldsymbol{v}) = 9$, so it follows from Theorem 5.6 that $\mathsf{q}_2(\lambda) = 9$.

### 5.4 Strategies for Problem 1.1 in the $k = 2$ case

We now relate these definitions and results back to Problem 1.1, in the case where only two positive query results are permitted. Recall as in Section 2.3 that we have the unknown ideal $\mu = \varnothing$ or $\mu = \lambda_{\preceq x}$ for some $x \in \lambda$. The $\lambda$-strategy $\boldsymbol{u} = (u_1,\ldots,u_r)$ defines a search strategy for $\mu$ as follows.

We query the elements $u_1, u_2, \ldots$ in sequence, until we have a positive query. If all the queries are negative, then, since $\lambda_{\succeq \boldsymbol{u}} = \lambda$, we have that $\mu = \varnothing$, and we are done

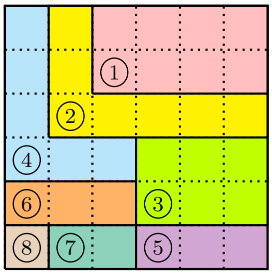



**Figure 7:** The poset $\lambda = [\![6,6]\!]$ with $\lambda$-strategy $\boldsymbol{v}$.

after $r \leq |\lambda_{\boldsymbol{u}}^{(r)}| + r - 1$ queries. Assume the query of $u_t$ is positive. Then the element $x$ is known to belong to $\lambda_{\succeq u_t}$, and known to not belong to $\lambda_{\succeq \{u_1,\ldots,u_{t-1}\}}$. Thus $x$ may be any of the elements in $\lambda_{\boldsymbol{u}}^{(t)}$. With one positive query remaining, the elements in $\lambda_{\boldsymbol{u}}^{(t)} \backslash \{u_t\}$ must be sequentially queried in any non-increasing order, as in the proof of Proposition 2.2. Thus, when the $u_t$ query is positive, $|\lambda_{\boldsymbol{u}}^{(t)}| + t - 1$ total queries are necessary to guarantee identification of $\mu$.

Therefore, by Definition 5.2, the value $\mathsf{Q}_2(\lambda, \boldsymbol{u})$ represents the maximum number of queries necessary to identify $\mu$ via the search strategy defined by $\boldsymbol{u}$. Thus, in view of Theorem 5.6, we may reframe the $k = 2$ case of Problem 1.1 in this combinatorial language:

**Problem 1, $k = 2$.** *Find the value $\mathsf{q}_2(\lambda)$, and identify a $\lambda$-strategy $\boldsymbol{u}$ such that $\mathsf{Q}_2(\lambda, \boldsymbol{u}) = \mathsf{q}_2(\lambda)$.*

## 6 Product posets of finite totally ordered sets

If $u = (a, b) \in \mathbb{N}^2$, we define the *transpose* element $u^T := (b, a)$. We extend this definition to sequences of elements $\boldsymbol{u} = (u_1, \ldots, u_r)$ in $\mathbb{N}^2$ and subsets $S \subset \mathbb{N}^2$ by setting:

$$\boldsymbol{u}^T := (u_1^T, \ldots, u_r^T), \qquad S^T := \{s^T \mid s \in S\}$$

The transpose map induces an isomorphism of posets $[\![m,n]\!] \cong [\![n,m]\!]$, for all $m, n \in \mathbb{N}$.

In this section it will be convenient to make use of a horizontally compressed visual shorthand for sequences of elements $\boldsymbol{v} = (v_1, \ldots, v_r)$ in $\lambda = [\![m,n]\!]$. Using the 'box array' representation of $[\![m,n]\!]$, we will label the element $v_i$ with ⓘ as usual, and then label every row in $\lambda_{\boldsymbol{v}}^{(i)}$ with the number of elements in that row. This visual information is sufficient to describe exactly all elements $v_i$ in $\boldsymbol{v}$, and the related sets $\lambda_{\boldsymbol{u}}^{(i)}$.

**Example 6.1.** Let $\lambda = [\![3,17]\!]$. If $\boldsymbol{v} = ((3,9),(2,13),(2,6),(1,15),(3,2),(1,4))$, then in Figure 8 we have the explicit visual representation of $\boldsymbol{v}$ (on the left) and the compressed shorthand representation of $\boldsymbol{v}$ (on the right).

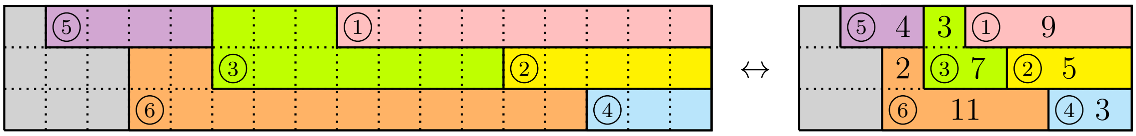



**Figure 8:** Compressed visual representation of the $\lambda$-strategy $\boldsymbol{v}$.

Now we prove the second main theorem of this paper.

**Theorem 6.1.** *Let $\lambda = [\![m, n]\!]$, with $m \leq 6$ or $n \leq 6$. Then we have:*

$$\mathsf{q}_2(\lambda) = \begin{cases} 9 & \text{if } m = n = 6; \\ \tau_2(mn) & \text{otherwise.} \end{cases}$$

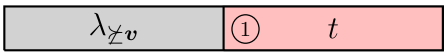
*Moreover, Algorithm 6.2 below produces an explicit $\lambda$-strategy $\boldsymbol{u}$ such that $\mathsf{Q}_2(\lambda, \boldsymbol{u}) = \mathsf{q}_2(\lambda)$.*

**Algorithm 6.2.** We assume $\lambda = [\![m, n]\!]$, with one of $m, n$ less than or equal to 6. This algorithm produces a $\lambda$-strategy $\boldsymbol{u}$ such that $\mathsf{Q}_2(\lambda, \boldsymbol{u}) = \mathsf{q}_2(\lambda)$.

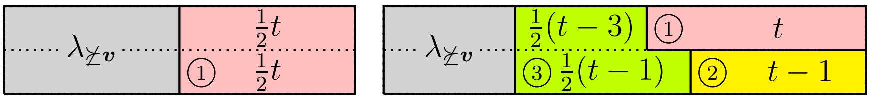
(`Step` 0) Let $\boldsymbol{u} = ()$ be the empty sequence. Go to (`Step` 1).

(`Step` 1) If the number of columns of $\lambda$ is greater than the number of rows, then redefine $\lambda := \lambda^T$, and set `flip` $= 1$. Otherwise set `flip` $= 0$. Redefine $m, n$ if necessary such that $\lambda = [\![m, n]\!]$. Go to (`Step` $m + 1$).

(`Step` 2, $\lambda = [\![1, n]\!]$). Define $t := \tau_2(n)$. Define $\boldsymbol{v}$ to be the one-element sequence in $\lambda$ depicted below. Go to (`Step` 8).

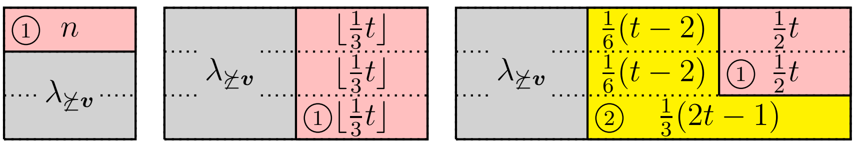



(`Step` 3, $\lambda = [\![2, n]\!]$). Define $t := \tau_2(2n)$. Define $\boldsymbol{v}$ to be the element sequence in $\lambda$ depicted below which corresponds to the appropriate condition on $t$. Go to (`Step` 8).

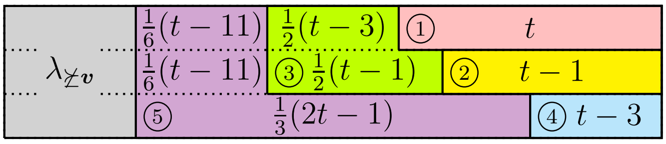



$t \equiv 0 \pmod 2$ $\qquad$ $t \equiv 1 \pmod 2$

(`Step` 4, $\lambda = [\![3, n]\!]$). Define $t := \tau_2(3n)$. Define $\boldsymbol{v}$ to be the element sequence in $\lambda$ depicted below which corresponds to the appropriate condition on $t$. Go to (`Step` 8).

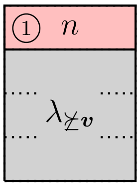



$t = 5$ $\qquad$ $t \equiv 0, 1 \pmod 3$ $\qquad$ $t \equiv 2 \pmod 6$

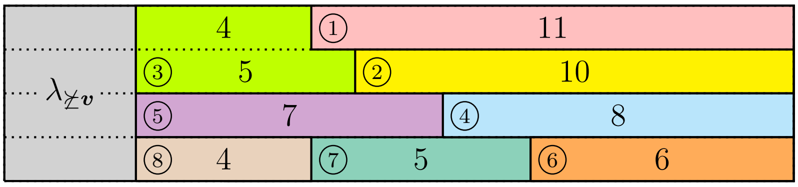



$11 \le t \equiv 5 \pmod 6$

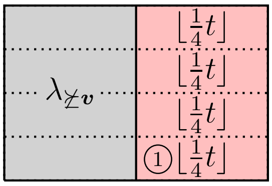
(`Step` 5, $\lambda = [\![4, n]\!]$). Define $t := \tau_2(4n)$. Define $\boldsymbol{v}$ to be the element sequence in $\lambda$ depicted below which corresponds to the appropriate condition on $t$. Go to (`Step` 8).



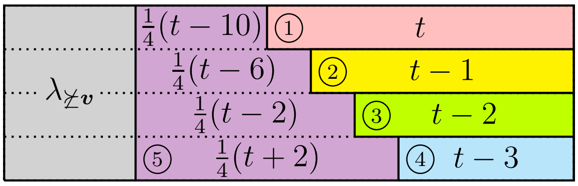

$t = 6, 7$



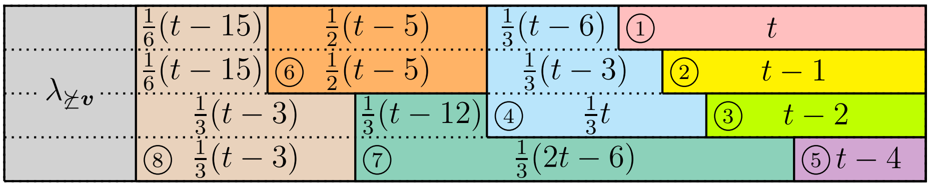

$t = 11$



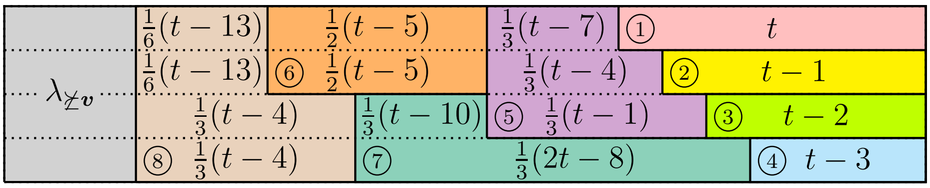

$t \equiv 0, 1 \pmod 4$



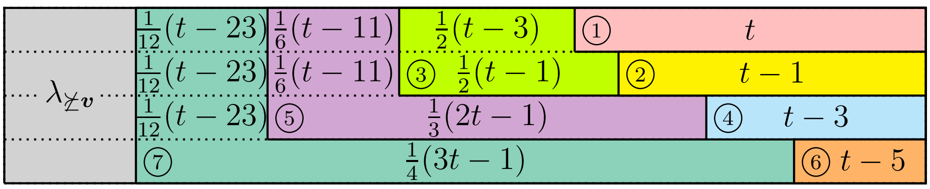

$10 \le t \equiv 2 \pmod 4$




$t \equiv 3 \pmod{12}$

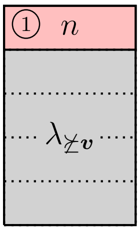



$19 \le t \equiv 7 \pmod{12}$

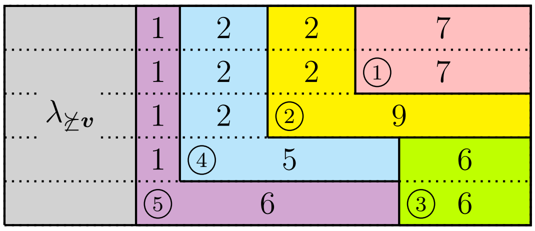



$23 \le t \equiv 11 \pmod{12}$

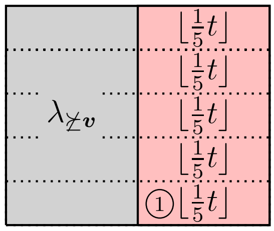
(Step 6, $\lambda = [\![5, n]\!]$). Define $t := \tau_2(5n)$. Define $\boldsymbol{v}$ to be the element sequence in $\lambda$ depicted below which corresponds to the appropriate condition on $t$. Go to (Step 8).



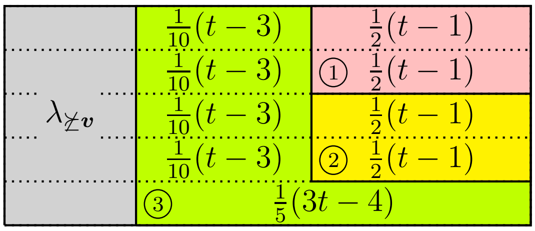

$t = 9$



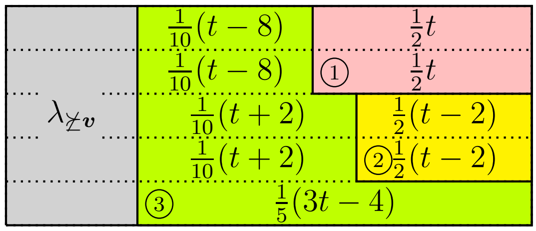

$t = 14$



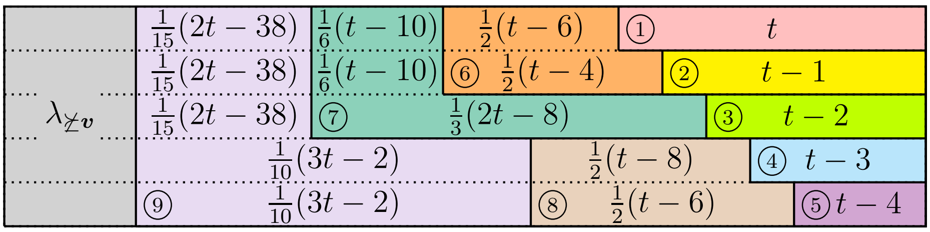

$t \equiv 0, 1, 2 \pmod 5$



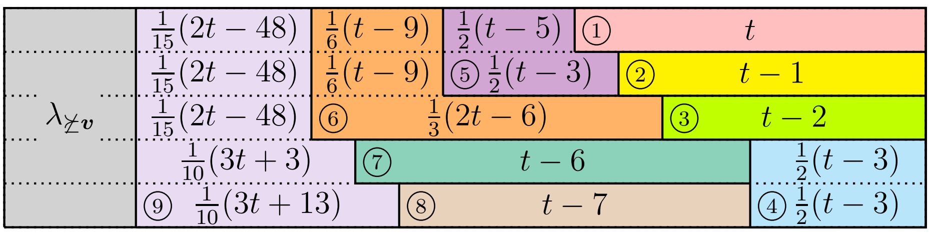

$t \equiv 3 \pmod{10}$



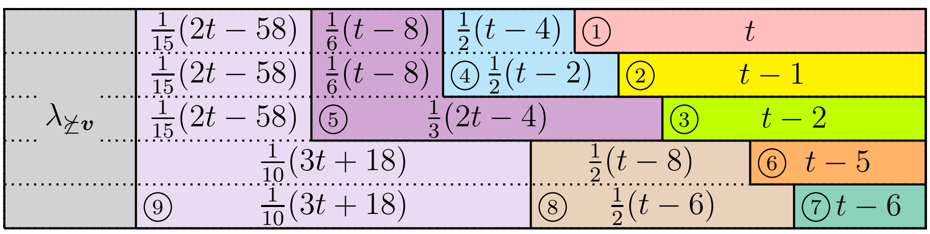

$t \equiv 8 \pmod{10}$



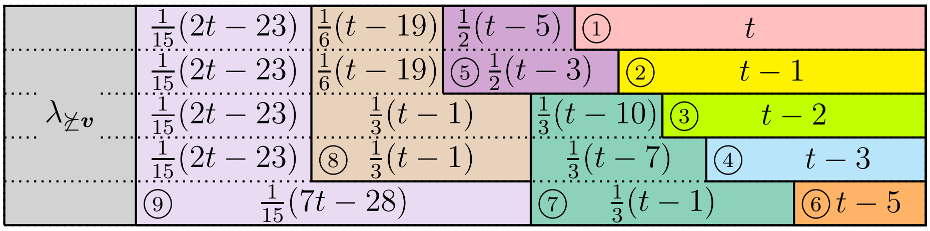

$34 \le t \equiv 4 \pmod{30}$



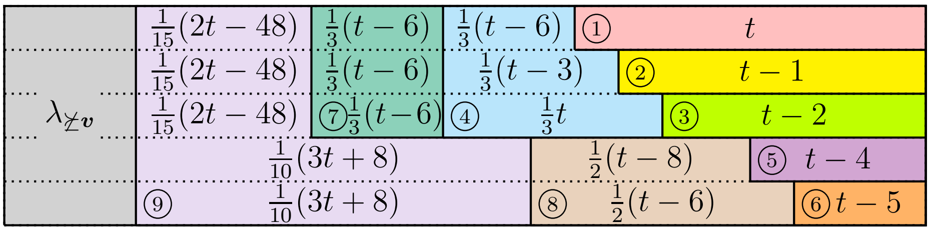

$39 \le t \equiv 9 \pmod{30}$



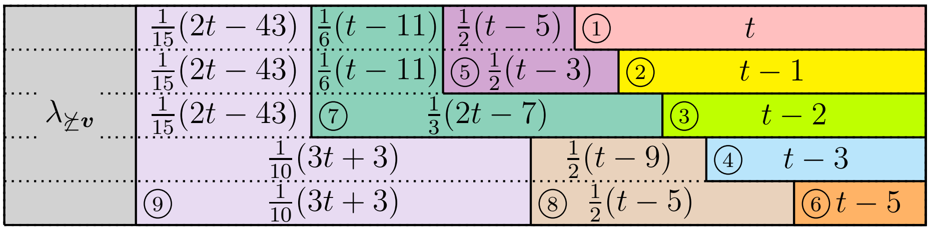

$44 \le t \equiv 14 \pmod{30}$



$t \equiv 19 \pmod{30}$

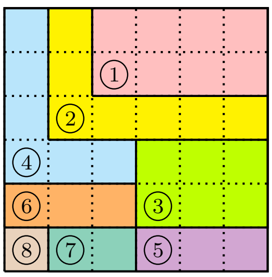


$t \equiv 24 \pmod{30}$

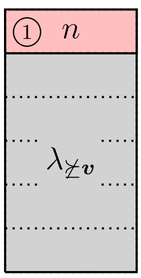


$t \equiv 29 \pmod{30}$

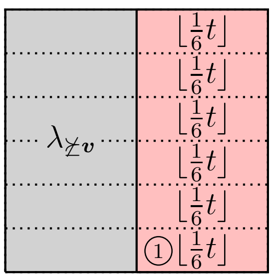
(`Step` 7, $\lambda = [\![6, n]\!]$). Define $t := \tau_2(6n)$. Define $\boldsymbol{v}$ to be the element sequence in $\lambda$ depicted below which corresponds to the appropriate conditions on $n$ and $t$. Go to (`Step` 8).



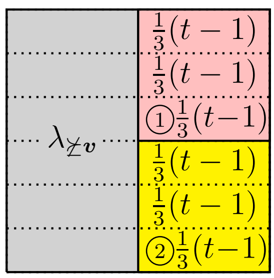
$n = 6$



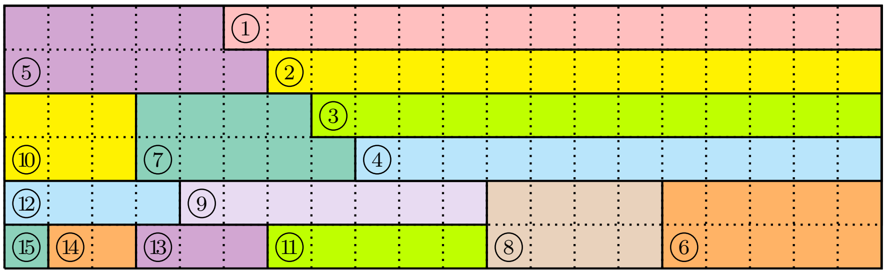
$7 \leq n \leq 11$



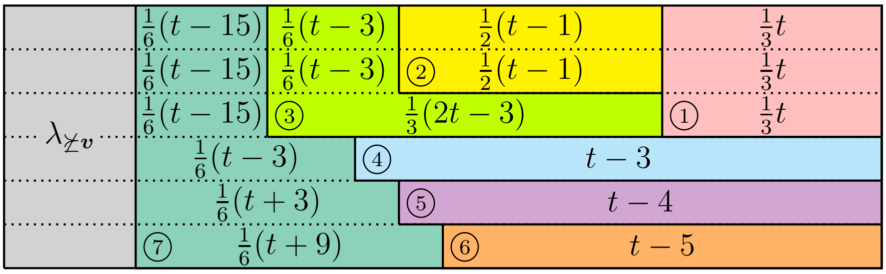
$n \in \{16,17,18,19,23,24\}$, or $t \equiv 0, 1 \pmod 6$



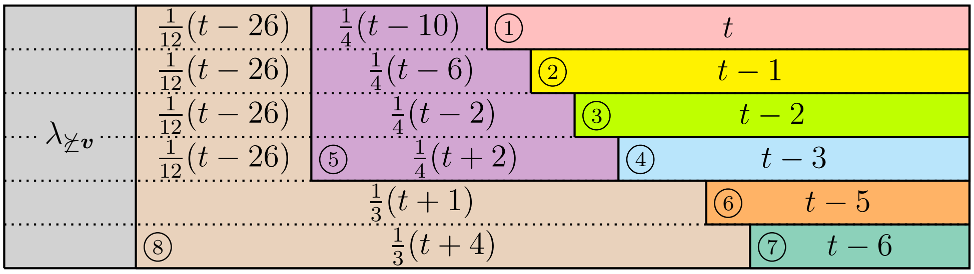
$16 \leq t \equiv 4 \pmod 6$



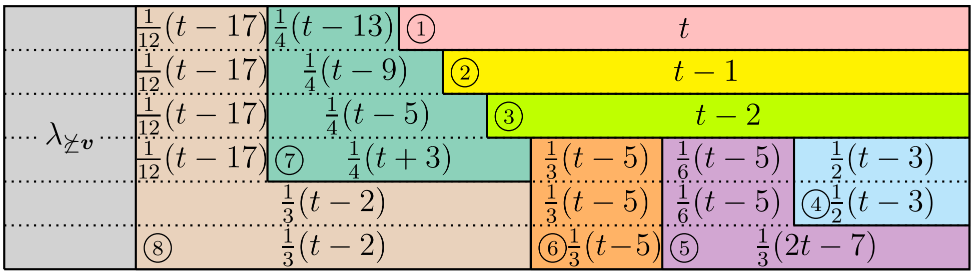
$n = 20$



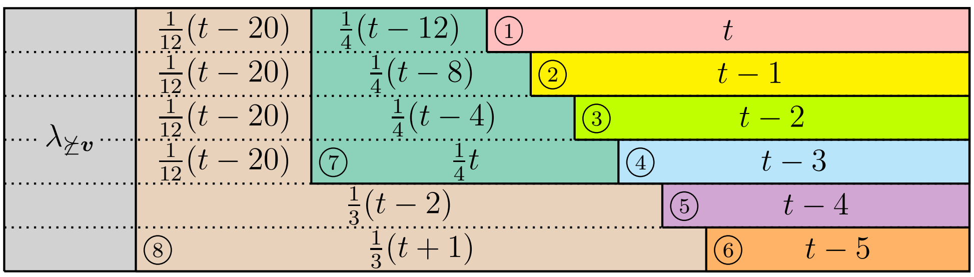
$21 \leq t \equiv 3 \pmod 6$



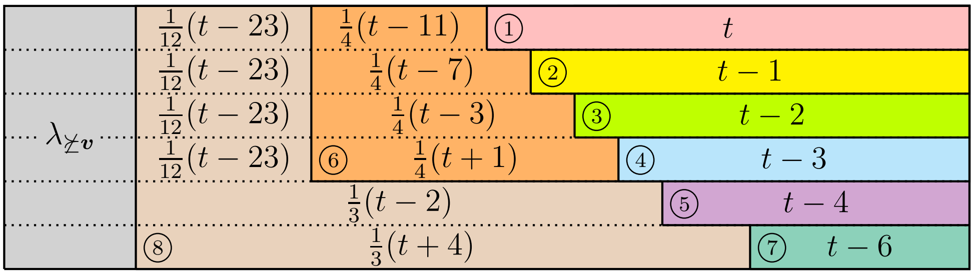
$26 \leq t \equiv 2 \pmod{12}$



$17 \leq t \equiv 5 \pmod{12}\,, n \geq 25$



$20 \leq t \equiv 8 \pmod{12}$




$23 \le t \equiv 11 \pmod{12}$

(`Step` 8) Set $\lambda' = \lambda_{\not\succeq \boldsymbol{v}}$. If `flip` $= 1$, set $\boldsymbol{v} := \boldsymbol{v}^T$ and $\lambda' := (\lambda')^T$. Redefine $\boldsymbol{u}$ to be the concatenation $\boldsymbol{uv}$. Go to (`Step` 9).

(`Step` 9) If $\lambda' = \varnothing$, END and return $\boldsymbol{u}$. Otherwise, redefine $\lambda := \lambda'$ and go to (`Step` 1).

*Proof.* First, one must check that the algorithm is well-defined; this entails verifying that the diagrams depicted in (`Steps` 2–7) describe a valid element sequence $\boldsymbol{v}$ (in particular, that the row labels are non-negative integers), and rests on the modular conditions for $t$ below each diagram. This is a straightforward exercise, and is left to the reader.

To begin, we consider the case $\lambda = [\![6, 6]\!]$. As discussed in Example 5.6, an exhaustive check shows that $\mathsf{Q}_2(\lambda, \boldsymbol{w}) > 8$ for all $\lambda$-strategies $\boldsymbol{w}$, so $\mathsf{q}_2(\lambda) > 8$. The $\lambda$-strategy defined in (`Step` 7) of the algorithm yields $8 < \mathsf{q}_2(\lambda) \le \mathsf{Q}_2(\lambda, \boldsymbol{u}) = 9$, so we have $\mathsf{q}_2(\lambda) = \mathsf{Q}_2(\lambda, \boldsymbol{u}) = 9$, as desired.

With that special case out of the way, we now prove, for all other diagrams under consideration, that Algorithm 6.2 produces a $\lambda$-strategy $\boldsymbol{u}$ such that $\mathsf{q}_2(\lambda) = \mathsf{Q}_2(\lambda, \boldsymbol{u}) = \tau_2(|\lambda|)$. We go by induction on $|\lambda|$. The base case $\lambda = [\![1, 1]\!]$ is clear, as the algorithm produces $\boldsymbol{u} = ((1, 1))$, and so $\mathsf{Q}_2(\lambda, \boldsymbol{u}) = \mathsf{q}_2(\lambda) = 1$.

Now let $\lambda = [\![m, n]\!]$, where $m \le 6$ or $n \le 6$, and $m, n$ are not both 6. Make the induction assumption that, if $\nu$ satisfies these conditions as well, with $|\nu| < |\lambda|$, then Algorithm 6.2 produces a $\nu$-strategy $\boldsymbol{w}$ such that $\mathsf{q}_2(\nu) = \mathsf{Q}_2(\nu, \boldsymbol{w}) = \tau_2(|\nu|)$.

Via the transpose operations in (`Steps` 1,8), it is enough to consider only the 'horizontally-oriented' situation $m \le n$, so we make that additional assumption now. We insert $\lambda = [\![m, n]\!]$ into Algorithm 6.2, letting $t = \tau_2(|\lambda|) = \tau_2(mn)$, and letting the element sequence $\boldsymbol{v} = (v_1, \dots, v_s)$ be as it stands at the end of (`Step` 8) in the first loop of the algorithm. We begin by arguing that $\lambda, \boldsymbol{v}$ satisfy the following three conditions:

(C1) $\mathsf{Q}_2(\lambda, \boldsymbol{v}) \le t$.

(C2) $|\lambda_{\succeq \boldsymbol{v}}| + T_2(t) - |\lambda| \ge st - T_2(s-1)$

(C3) $\lambda_{\not\succeq \boldsymbol{v}} \ne [\![6, 6]\!]$.

First we check that (C1) is satisfied by considering every diagram in (`Steps` 2–7), save for the $[\![6,6]\!]$ diagram. The homogeneously-colored component of the diagram marked with the element ⓘ in the southwest corner is exactly the set $\lambda_{\boldsymbol{v}}^{(i)}$. By adding up the elements in each row of $\lambda_{\boldsymbol{v}}^{(i)}$, it is straightforward to check that in all cases, we have $|\lambda_{\boldsymbol{v}}^{(i)}| \leq t-i+1$. Then we have:

$$\begin{aligned}\mathsf{Q}_2(\lambda,\boldsymbol{v}) &= \max\{|\lambda_{\boldsymbol{v}}^{(i)}|+i-1 \mid i=1,\dots,s\}\\ &\leq \max\{(t-i+1)+i-1 \mid i=1,\dots,s\}=t.\end{aligned}$$

Now we check that $\lambda, \boldsymbol{v}$ satisfy (C2) by considering every diagram in (`Steps` 2–7), save for the $[\![6,6]\!]$ diagram. We do so in the separate Cases 1–7 below.

(Case 1) Consider the small cases of the form:

- $m=3$, $n\in\{4,5\}$ (and so $t=5$)
- $m=4$, $n\in\{4,\dots,7\}$ (and so $t\in\{6,7\}$)
- $m=5$, $n\in\{6,7\}$ (and so $t=8$)
- $m=6$, $n\in\{7,\dots,11\}$ (and so $t\in\{9,\dots,11\}$)

In all these cases, we have $s=1$, and $|\lambda_{\succeq\boldsymbol{v}}|=n$. It is easily checked on a case-by-case basis that $T_2(t)-|\lambda|=T_2(t)-mn\geq t-n$, so we have

$$|\lambda_{\succeq\boldsymbol{v}}|+T_2(t)-|\lambda|\geq n+(t-n)=t=t-T_2(0),$$

satisfying (C2).

(Case 2) Consider the small cases of the form:

- $m=6$, $n\in\{16,17,18,19,23,24\}$ (and so $t\in\{14,15,17\}$)

In all these cases, we have $s=1$, and $|\lambda_{\succeq\boldsymbol{v}}|=6\cdot\lfloor\frac{t}{6}\rfloor=12$. It is easily checked on a case-by-case basis that $12+T_2(t)-6n\geq t$, so we have

$$|\lambda_{\succeq\boldsymbol{v}}|+T_2(t)-|\lambda|=12+T_2(t)-6n\geq t=t-T_2(0),$$

satisfying (C2).

(Case 3) Consider the case $\lambda=[\![m,n]\!]$, where $3\leq m\leq 6$, and $t\equiv 1 \pmod{m}$, as in (`Steps` 4,5,6,7). Then $s=1$, and $T_2(t)\not\equiv 0 \pmod{m}$ by Lemma 3.11. We also have $|\lambda|\equiv 0 \pmod{m}$, so $T_2(t)-|\lambda|\geq 1$ follows by Lemma 3.8. Therefore

$$|\lambda_{\succeq\boldsymbol{v}}|+T_2(t)-|\lambda|\geq|\lambda_{\succeq\boldsymbol{v}}|+1=m\left\lfloor\frac{t}{m}\right\rfloor+1=m\cdot\frac{t-1}{m}+1=t=t-T_2(0),$$

satisfying (C2).

(Case 4) Consider the case $\lambda = [\![5, n]\!]$ and $t \equiv 2 \pmod 5$, as in (`Step` 6). Then $s = 1$, and $T_2(t) \equiv T_2(2) \equiv 3 \pmod 5$ by Lemma 3.11. We also have $|\lambda| \equiv 0 \pmod 5$, so $T_2(t) - |\lambda| \geq 3$ follows by Lemma 3.8. Therefore

$$|\lambda_{\succeq \boldsymbol{v}}| + T_2(t) - |\lambda| \geq |\lambda_{\succeq \boldsymbol{v}}| + 3 = 5 \left\lfloor \frac{t}{5} \right\rfloor + 3 = 5 \cdot \frac{t-2}{5} + 3 = t + 1 > t = t - T_2(0),$$

satisfying (C2).

(Case 5) Consider the case $\lambda = [\![5, n]\!]$ and $t \equiv 3 \pmod 5$. Then we have $t \equiv 3 \pmod{10}$ or $t \equiv 8 \pmod{10}$ as in (`Step` 6). Then in either case $s = 3$, and $T_2(t) \equiv T_2(3) \equiv 1 \pmod 5$ by Lemma 3.11. We also have $|\lambda| \equiv 0 \pmod 5$, so $T_2(t) - |\lambda| \geq 1$ follows by Lemma 3.8. Therefore

$$|\lambda_{\succeq \boldsymbol{v}}| + T_2(t) - |\lambda| \geq |\lambda_{\succeq \boldsymbol{v}}| + 1 = (3t - 4) + 1 = 3t - 3 = 3t - T_2(2),$$

satisfying (C2).

(Case 6) Consider the case $\lambda = [\![6, n]\!]$ and $16 \leq t \equiv 4 \pmod 6$, as in (`Step` 7). Then $s = 2$, and $T_2(t) \not\equiv 0 \pmod 6$ by Lemma 3.11. We also have $|\lambda| \equiv 0 \pmod 6$, so $T_2(t) - |\lambda| \geq 1$ follows by Lemma 3.8. Therefore

$$|\lambda_{\succeq \boldsymbol{v}}| + T_2(t) - |\lambda| \geq |\lambda_{\succeq \boldsymbol{v}}| + 1 = (2t - 2) + 1 = 2t - 1 = 2t - T_2(1),$$

satisfying (C2).

(Case 7) Now we may consider the remaining cases in one fell swoop. In all remaining cases, it may be checked that $\boldsymbol{v}$ is defined such that $|\lambda_{\boldsymbol{v}}^{(i)}| = t - i + 1$ for $t = 1, \ldots, s$, and thus $|\lambda_{\succeq \boldsymbol{v}}| = st - T_2(s-1)$. Therefore we have

$$|\lambda_{\succeq \boldsymbol{v}}| + T_2(t) - |\lambda| \geq |\lambda_{\succeq \boldsymbol{v}}| = st - T_2(s-1),$$

satisfying (C2).

Now we check that $\lambda, \boldsymbol{v}$ satisfy (C3). The case $\lambda = [\![m, n]\!]$ for $m < 6$ is obvious. Thus we may assume that $\lambda = [\![6, n]\!]$. As with the last claim, we check (C3) in the separate Cases 1–10 below.

(Case 1) If $7 \leq n \leq 11$, then $\lambda_{\not\succeq \boldsymbol{v}}$ is a 5-row diagram, so is not equal to $[\![6, 6]\!]$.

(Case 2) If $12 \leq n \leq 15$, then $t \in \{12, 13\}$, so $t \equiv 0, 1 \pmod 6$. Then $\lfloor \frac{t}{6} \rfloor = 2$, and we have $|\lambda_{\not\succeq \boldsymbol{v}}| = |\lambda| - 6 \cdot \lfloor \frac{t}{6} \rfloor \geq 12 \cdot 6 - 12 = 60$, so $\lambda_{\not\succeq \boldsymbol{v}} \neq [\![6, 6]\!]$.

(Case 3) If $n \in \{16, 17, 18, 19, 23, 24\}$, then $t \in \{14, 15, 17\}$, so $\lfloor \frac{t}{6} \rfloor = 2$, and we have $|\lambda_{\not\succeq \boldsymbol{v}}| = |\lambda| - 6 \cdot \lfloor \frac{t}{6} \rfloor \geq 16 \cdot 6 - 12 = 84$, so $\lambda_{\not\succeq \boldsymbol{v}} \neq [\![6, 6]\!]$.

(Case 4) If $n = 20$, then $\lambda_{\not\succeq \boldsymbol{v}} = \varnothing \neq [\![6, 6]\!]$.

(Case 5) If $n \in \{21, 22\}$, then $t = 16$. Then $|\lambda_{\not\succeq \boldsymbol{v}}| = |\lambda| - (2t - 2) \geq (21 \cdot 6) - 30 = 96$, so $\lambda_{\not\succeq \boldsymbol{v}} \neq [\![6, 6]\!]$.

(Case 6) If $n = 25$, then $t = 17$. Then $|\lambda_{\not\succeq \boldsymbol{v}}| = |\lambda| - (8t - 28) = (25 \cdot 6) - (8 \cdot 17 - 28) = 42$, so $\lambda_{\not\succeq \boldsymbol{v}} \neq [\![6, 6]\!]$.

In the remaining cases, we assume that $n \geq 26$. Then we have $t \geq 18$. Note that by the definition of $t = \tau_2(|\lambda|)$, we have $|\lambda| > T_2(t-1)$.

(Case 7) If $t \equiv 0, 1 \pmod 6$, then

$$\begin{aligned}|\lambda_{\not\geq \boldsymbol{v}}| &= |\lambda| - 6\left\lfloor \frac{t}{6} \right\rfloor > T_2(t-1) - t = T_2(t-1) - (t-1) - 1\\ &= T_2(t-2) - 1 \geq T_2(16) - 1 = 135,\end{aligned}$$

so $\lambda_{\not\geq \boldsymbol{v}} \neq [\![6, 6]\!]$.

(Case 8) Say $t \equiv 4 \pmod 6$. Then

$$\begin{aligned}|\lambda_{\not\geq \boldsymbol{v}}| &= |\lambda| - (2t-2) > T_2(t-1) - (2t-2)\\ &= T_2(t-1) - (t-1) - (t-2) - 1 = T_2(t-3) - 1 \geq T_2(15) - 1 = 119,\end{aligned}$$

so $\lambda_{\not\geq \boldsymbol{v}} \neq [\![6, 6]\!]$.

(Case 9) Say $t \equiv 3 \pmod 6$. Then

$$\begin{aligned}|\lambda_{\not\geq \boldsymbol{v}}| &= |\lambda| - (7t-21) > T_2(t-1) - (t-1) - \cdots - (t-7) - 7\\ &= T_2(t-8) - 7 \geq T_2(10) - 7 = 48,\end{aligned}$$

so $\lambda_{\not\geq \boldsymbol{v}} \neq [\![6, 6]\!]$.

(Case 10) Say $t \equiv 2, 5 \pmod 6$. Then

$$\begin{aligned}|\lambda_{\not\geq \boldsymbol{v}}| &= |\lambda| - (8t-28) > T_2(t-1) - (t-1) - \cdots - (t-8) - 8\\ &= T_2(t-9) - 8 \geq T_2(9) - 8 = 37,\end{aligned}$$

so $\lambda_{\not\geq \boldsymbol{v}} \neq [\![6, 6]\!]$.

Thus, in every case we have $\lambda_{\not\geq \boldsymbol{v}} \neq [\![6, 6]\!]$, and so (C3) holds.

Therefore (C1), (C2), (C3) hold for $\lambda, \boldsymbol{v}$. By (C2), we have

$$|\lambda_{\not\geq \boldsymbol{v}}| = |\lambda| - |\lambda_{\geq \boldsymbol{v}}| \leq T_2(\tau_2(|\lambda|)) - s\tau_2(|\lambda|) + T_2(s-1),$$

so by Lemmas 3.5 and 3.10 we have

$$\tau_2(|\lambda_{\not\geq \boldsymbol{v}}|) \leq \tau_2(T_2(\tau_2(|\lambda|)) - s\tau_2(|\lambda|) + T_2(s-1)) \leq \tau_2(|\lambda|) - s. \tag{6.3}$$

By (C3), the induction assumption holds for $\lambda_{\not\geq \boldsymbol{v}}$, so inserting $\lambda_{\not\geq \boldsymbol{v}}$ into the algorithm yields a $\lambda_{\not\geq \boldsymbol{v}}$-strategy $\boldsymbol{w}$ such that $\mathsf{Q}_2(\lambda_{\not\geq \boldsymbol{v}}, \boldsymbol{w}) = \mathsf{q}_2(\lambda_{\not\geq \boldsymbol{v}}) = \tau_2(|\lambda_{\not\geq \boldsymbol{v}}|)$. By the inductive nature of the algorithm, inserting $\lambda$ into the algorithm yields the $\lambda$-strategy $\boldsymbol{u} = \boldsymbol{v}\boldsymbol{w}$. Then we have

$$\begin{aligned}\mathsf{Q}_2(\lambda, \boldsymbol{u}) &= \max\{\mathsf{Q}_2(\lambda, \boldsymbol{v}), \mathsf{Q}_2(\lambda_{\not\geq \boldsymbol{v}}, \boldsymbol{w}) + s\} && \text{by Lemma 5.5}\\ &\leq \max\{\tau_2(|\lambda|), \mathsf{Q}_2(\lambda_{\not\geq \boldsymbol{v}}, \boldsymbol{w}) + s\} && \text{by (C1)}\\ &= \max\{\tau_2(|\lambda|), \tau_2(|\lambda_{\not\geq \boldsymbol{v}}|) + s\} && \text{by induction assumption}\\ &\leq \max\{\tau_2(|\lambda|), (\tau_2(|\lambda|) - s) + s\} && \text{by (6.3)}\\ &= \tau_2(|\lambda|).\end{aligned}$$

Then, by Theorems 4.2 and 5.6, we have

$$\tau_2(|\lambda|) \leq \mathsf{q}_2(\lambda) \leq \mathsf{Q}_2(\lambda, \boldsymbol{u}) \leq \tau_2(|\lambda|),$$

so we have $\mathsf{q}_2(\lambda) = \mathsf{Q}_2(\lambda, \boldsymbol{u}) = \tau_2(|\lambda|)$ as desired, completing the proof. $\square$

As $\kappa \times \nu \cong [\![m, n]\!]$ when $\kappa$, $\nu$ are totally ordered sets of cardinality $m, n$ respectively, we have the immediate corollary thanks to Lemma 4.1:

**Corollary 6.2.** *Let $\kappa, \nu$ be finite totally ordered sets, with $|\kappa| \leq 6$ or $|\nu| \leq 6$. Then*

$$\mathsf{q}_2(\kappa \times \nu) = \begin{cases} 9 & \text{if } |\kappa| = |\nu| = 6; \\ \tau_2(|\kappa||\nu|) & \text{otherwise.} \end{cases}$$

### 6.1 A conjecture

We end with a conjectural bound for product posets of totally ordered sets.

**Conjecture 6.3.** *Let $m, n \in \mathbb{N}$. Then $\mathsf{q}_2([\![m, n]\!]) \leq \tau_2(mn) + 1$.*

This suggests $\mathsf{q}_2([\![m, n]\!]) \in \{\tau_2(mn), \tau_2(mn) + 1\}$ for all $m, n \in \mathbb{N}$. By Theorem 6.1, the posets $[\![m, n]\!]$ obey this claim when $m \leq 6$ or $n \leq 6$. In fact, all but $[\![6, 6]\!]$ have the minimal possible value $\mathsf{q}_2([\![m, n]\!]) = \tau_2(mn)$ allowed by Theorem 4.2. Moving beyond these results, computations show that exceptional cases like $[\![6, 6]\!]$, where no $\lambda$-strategy $\boldsymbol{u}$ can be found that realizes $\mathsf{Q}_2(\lambda, \boldsymbol{u}) = \tau_2(|\lambda|)$, seem to occur fairly rarely (the poset $[\![15, 20]\!]$ is another). But allowing for a $\lambda$-strategy that realizes $\mathsf{Q}_2(\lambda, \boldsymbol{u}) = \tau_2(|\lambda|) + 1$ instead seems to afford so much flexibility that we expect such a $\lambda$-strategy can always be found, even in these exceptional cases. For instance, while there are no $[\![6, 6]\!]$-strategies that realize $\mathsf{Q}_2([\![6, 6]\!], \boldsymbol{u}) = 8$, there are 53,688 distinct $[\![6, 6]\!]$-strategies which realize $\mathsf{Q}_2([\![6, 6]\!], \boldsymbol{u}) = 9$. This is the authors' line of reasoning behind positing Conjecture 6.3.

## References


[1] S. Angelov, K. Kunal and A. McGregor, Sorting and Selection with Random Costs, *LATIN 2008: Theoretical Informatics*, (2008), 8–59.

[2] Y. Ben-Asher, E. Farchi and I. Newman, Optimal search in trees, *SIAM J. Comput.* **28**(6) (1999), 2090–2102.

[3] Y. Chen, An efficient search algorithm for partially ordered sets, *Proceedings of the 2nd IASTED international conference on advances in computer science and technology* (2006), 91–94.

[4] F. Cicalese, T. Jacobs, E. Laber and M. Molinaro, On the complexity of searching in trees and partially ordered structures, *Theor. Comput. Sci.* **412** (2011), 6879–6896.

[5] F. Cicalese, T. Jacobs, E. Laber and M. Molinaro, On Greedy Algorithms for Decision Trees, *ISAAC 2010: Algorithms and Computation*, (2010), 206–217.

[6] F. Cicalese, Balázs Keszegh, B. Lidický, D.Pálvölgyi and T. Valla, On the Tree Search Problem with Non-uniform Costs, *Theor. Comput. Sci.* **98**(1) (1999), 39–63.

[7] C. Daskalakis, R. M. Karp, E. Mossel, S. Riesenfeld and E. Verbin, Sorting and selection in posets, *ACM-SIAM Symposium on Discrete Algorithms*, (2009), 392–401.

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

[9] R. Denman, D. Hailey and M. Rothenberg, The Tower and Glass Marbles Problem, *College Math. J.* **41** 5 (2010), 350–356.

[10] D. Dereniowski, Edge ranking and searching in partial orders, *Discrete Appl. Math.* **156**(13), (2008), 2493–2500.

[11] B. Dushnik and E. Miller, Partially ordered sets, *Amer. J. Math.* **63** (1941), 600–610.

[12] J. D. E. Konhauser, D. Velleman and S. Wagon, (1996), *Which way did the Bicycle Go?* Dolciani Mathematical Expositions, No. 18, The Mathematical Association of America, 1996.

[13] G. L. McDowell, *Cracking the Coding Interview, 5th ed.*, CareerCup, 2011.

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

(Received 8 July 2020; revised 17 Dec 2020)