Faster Pattern Matching under Edit Distance

Panagiotis Charalampopoulos[1], Tomasz Kociumaka[2], Philip Wellnitz[2]

1. Birkbeck, University of London, UK

2. Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany

FOCS 2022

Denver, USA

The Problem

The Problem

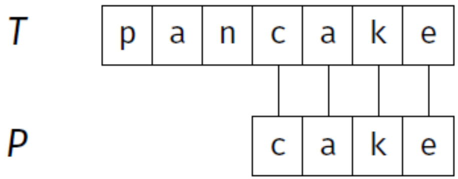
Pattern Matching

Given a text T and a pattern P, compute the occurrences of P in T.

The Problem

Pattern Matching

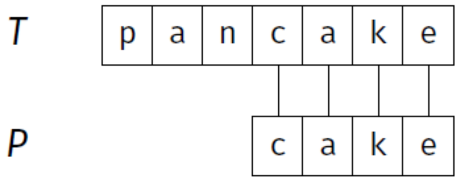
Given a text T and a pattern P, compute the occurrences of P in T.



The Problem

Pattern Matching

Given a text T and a pattern P, compute the occurrences of P in T.



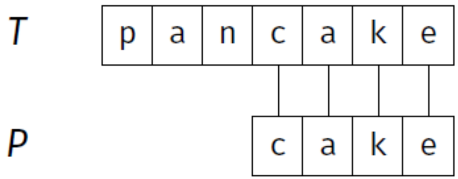
Pattern Matching under Edit Distance

Given a text T, a pattern P, and an integer threshold k, compute the (starting positions of) substrings of T that are at edit distance at most k from P.

The Problem

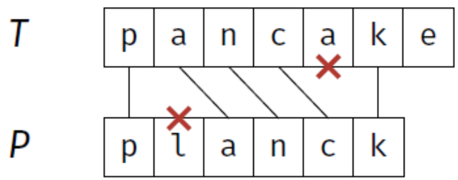
Pattern Matching

Given a text T and a pattern P, compute the occurrences of P in T.



Pattern Matching under Edit Distance

Given a text T, a pattern P, and an integer threshold k, compute the (starting positions of) substrings of T that are at edit distance at most k from P.



History and our Result

History and our Result

$\mathcal{O}(n^2)$ [Sellers; J. Algorithms 1980]

History and our Result

$\mathcal{O}(n^2)$	[Sellers; J. Algorithms 1980]
$\mathcal{O}(nk^2)$	[Landau, Vishkin; JCSS 1988]

History and our Result

$\mathcal{O}(n^2)$ [Sellers; J. Algorithms 1980]

$\mathcal{O}(nk^2)$ [Landau, Vishkin; JCSS 1988]

$\mathcal{O}(nk)$ [Landau, Vishkin; J. Algorithms 1989]

History and our Result

$\mathcal{O}(n^2)$	[Sellers; J. Algorithms 1980]
$\mathcal{O}(nk^2)$	[Landau, Vishkin; JCSS 1988]
$\mathcal{O}(nk)$	[Landau, Vishkin; J. Algorithms 1989]
$\tilde{\mathcal{O}}(n + k^{8+1/3} \cdot n/m^{1/3})$	[Sahinalp, Vishkin; FOCS 1996]

History and our Result

$\mathcal{O}(n^2)$	[Sellers; J. Algorithms 1980]
$\mathcal{O}(nk^2)$	[Landau, Vishkin; JCSS 1988]
$\mathcal{O}(nk)$	[Landau, Vishkin; J. Algorithms 1989]
$\tilde{\mathcal{O}}(n + k^{8+1/3} \cdot n/m^{1/3})$	[Sahinalp, Vishkin; FOCS 1996]
$\mathcal{O}(n + k^4 \cdot n/m)$	[Cole, Hariharan; SICOMP 2002]

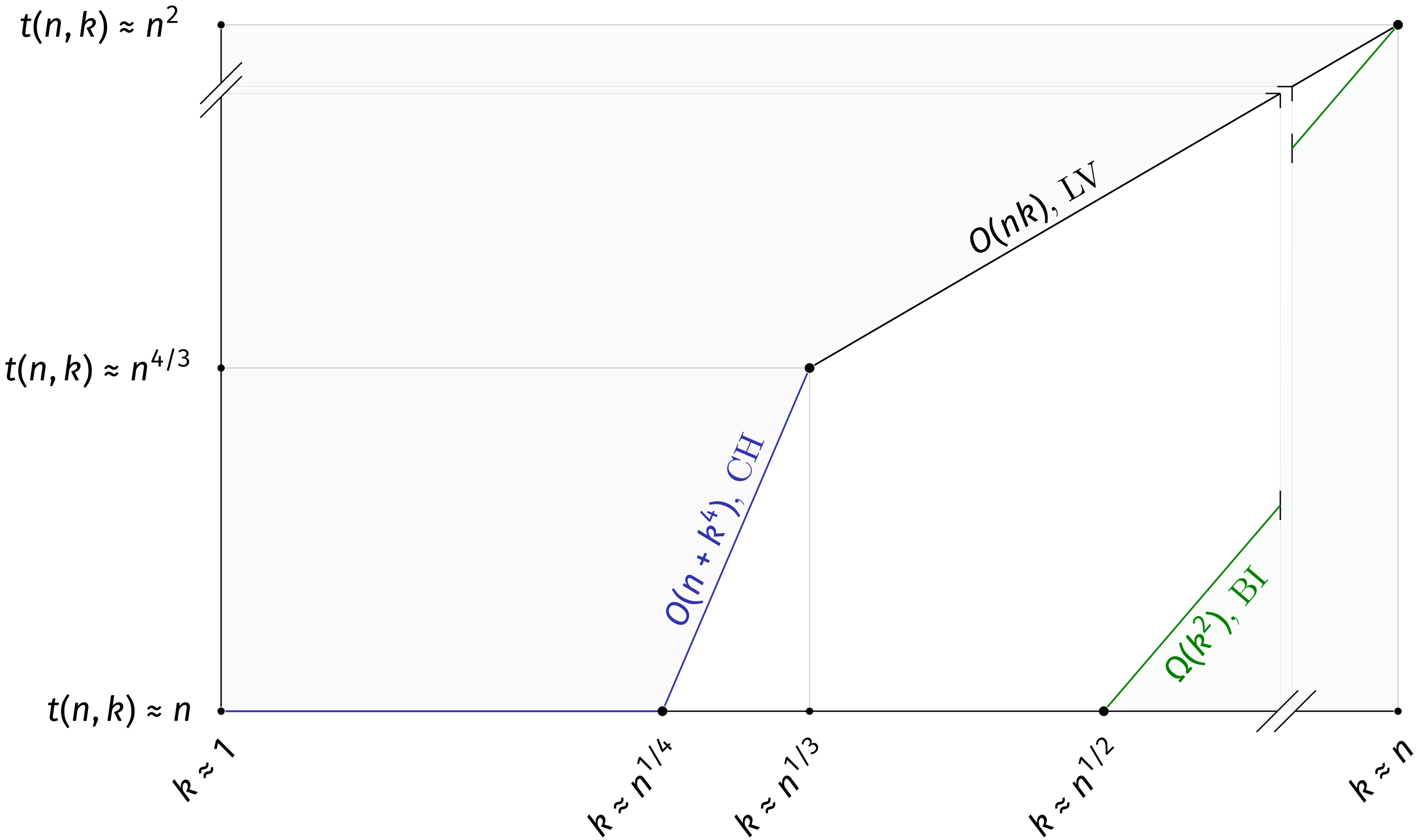
History and our Result

$\mathcal{O}(n^2)$	[Sellers; J. Algorithms 1980]
$\mathcal{O}(nk^2)$	[Landau, Vishkin; JCSS 1988]
$\mathcal{O}(nk)$	[Landau, Vishkin; J. Algorithms 1989]
$\tilde{\mathcal{O}}(n + k^{8+1/3} \cdot n/m^{1/3})$	[Sahinalp, Vishkin; FOCS 1996]
$\mathcal{O}(n + k^4 \cdot n/m)$	[Cole, Hariharan; SICOMP 2002]
$\tilde{\mathcal{O}}(n + k^{3.5} \cdot n/m)$	This work

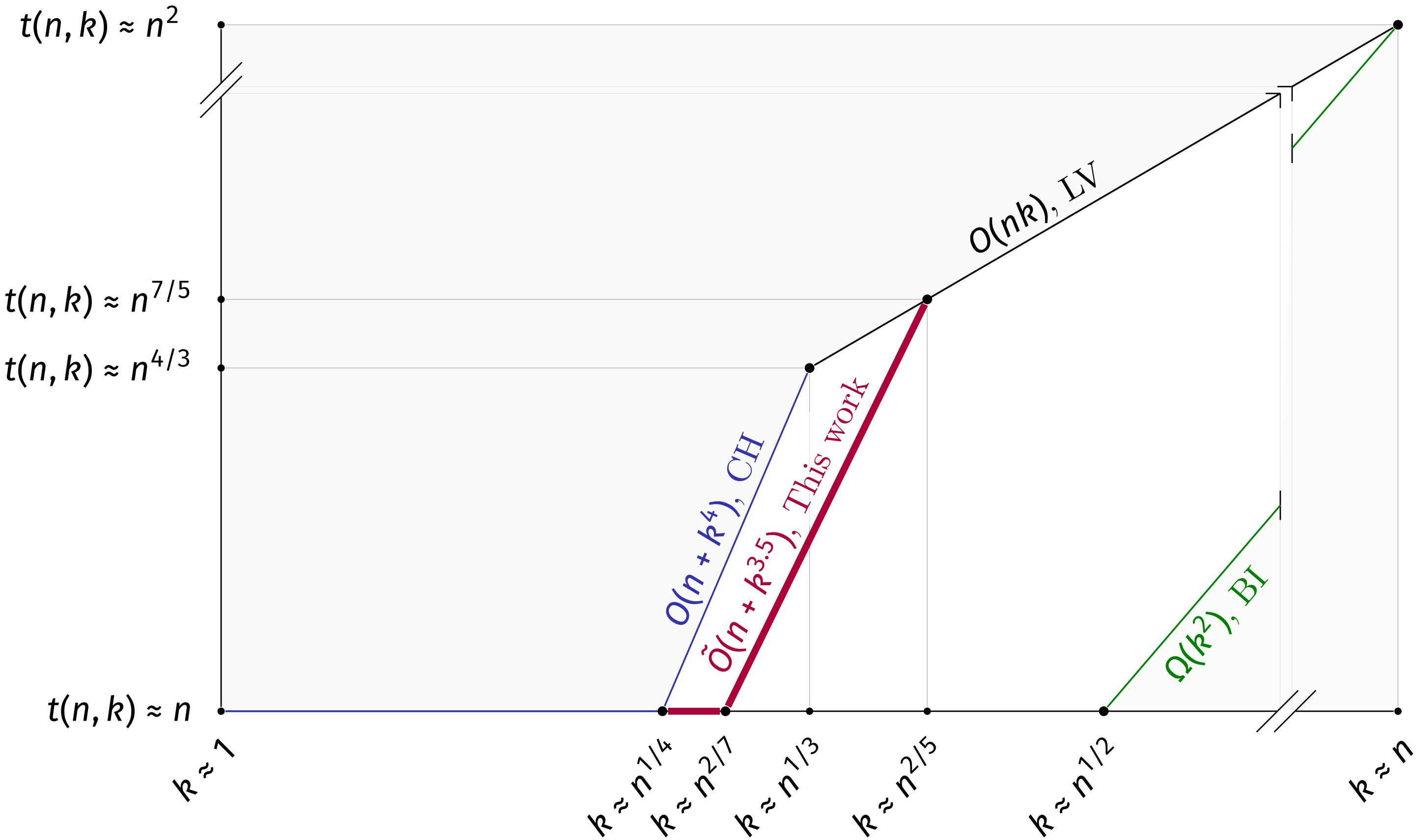
History and our Result

$\mathcal{O}(n^2)$	[Sellers; J. Algorithms 1980]
$\mathcal{O}(nk^2)$	[Landau, Vishkin; JCSS 1988]
$\mathcal{O}(nk)$	[Landau, Vishkin; J. Algorithms 1989]
$\tilde{\mathcal{O}}(n + k^{8+1/3} \cdot n/m^{1/3})$	[Sahinalp, Vishkin; FOCS 1996]
$\mathcal{O}(n + k^4 \cdot n/m)$	[Cole, Hariharan; SICOMP 2002]
$\tilde{\mathcal{O}}(n + k^{3.5} \cdot n/m)$	This work
$\Omega(k^2)$	[Backurs, Indyk; SICOMP 2018]

History and our Result

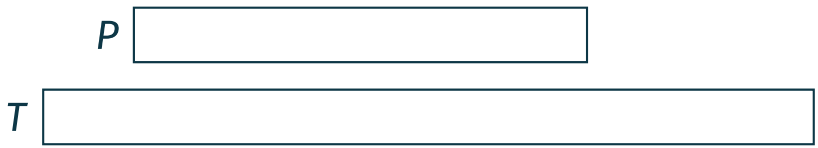


History and our Result



The Structure of Pattern Matching

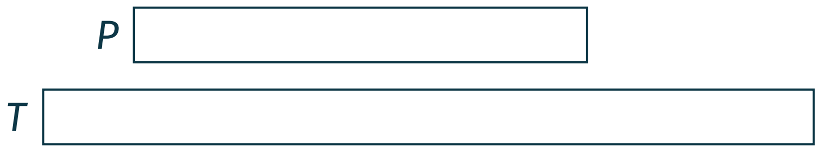
Fact [folklore] Given a pattern P of length m and a text T of length $n \leq {}^{3}\!/_{2}\, m$ at least one of the following holds:



The Structure of Pattern Matching

Fact [folklore] Given a pattern P of length m and a text T of length $n \leq 3/2\, m$ at least one of the following holds:

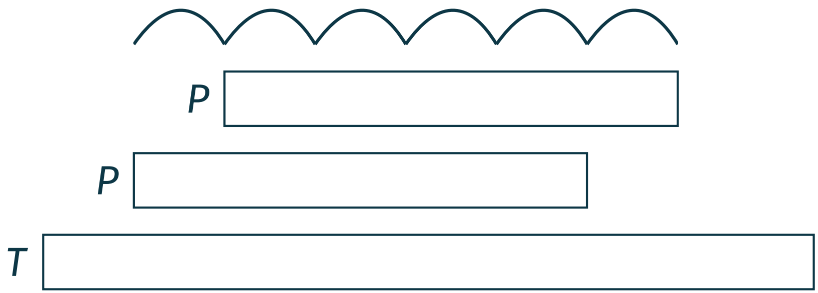
- The pattern P has at most one occurrence in T.



The Structure of Pattern Matching

Fact [folklore] Given a pattern P of length m and a text T of length $n \leq 3/2\, m$ at least one of the following holds:

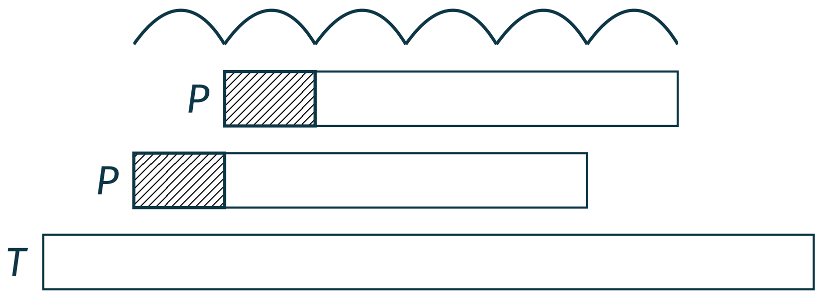
- The pattern P has at most one occurrence in T.
- The pattern P is periodic.



The Structure of Pattern Matching

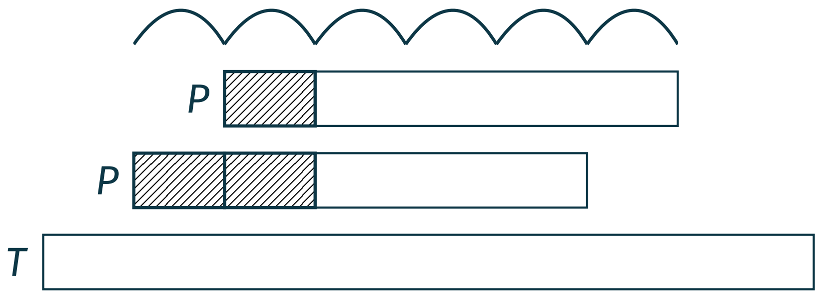
Fact [folklore] Given a pattern P of length m and a text T of length $n \leq {}^{3}\!/_{2}\, m$ at least one of the following holds:

- The pattern P has at most one occurrence in T.
- The pattern P is periodic.



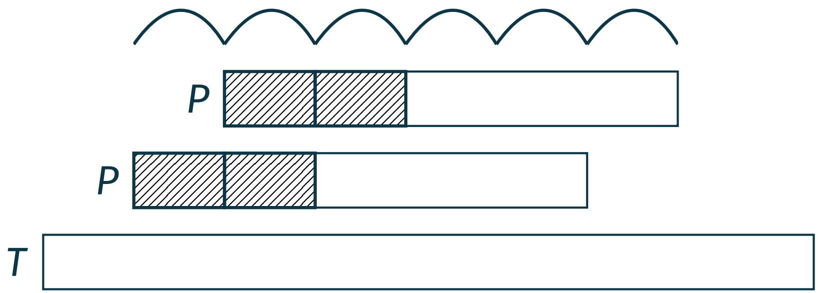
Fact [folklore] Given a pattern P of length m and a text T of length $n \leq {}^3\!/_2\, m$ at least one of the following holds:

- The pattern P has at most one occurrence in T.
- The pattern P is periodic.



Fact [folklore] Given a pattern P of length m and a text T of length $n \leq {}^{3}\!/_{2}\, m$ at least one of the following holds:

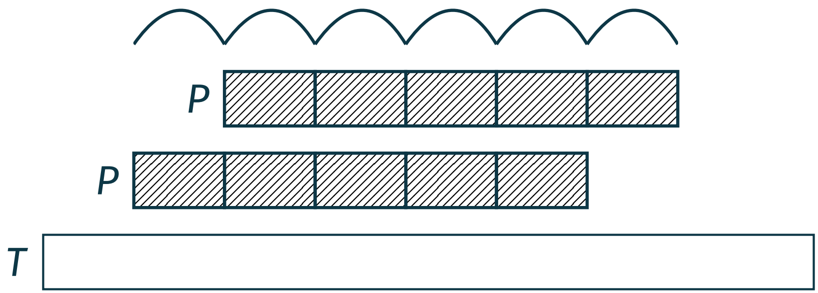
- The pattern P has at most one occurrence in T.
- The pattern P is periodic.



The Structure of Pattern Matching

Fact [folklore] Given a pattern P of length m and a text T of length $n \leq {}^{3}\!/_{2}\, m$ at least one of the following holds:

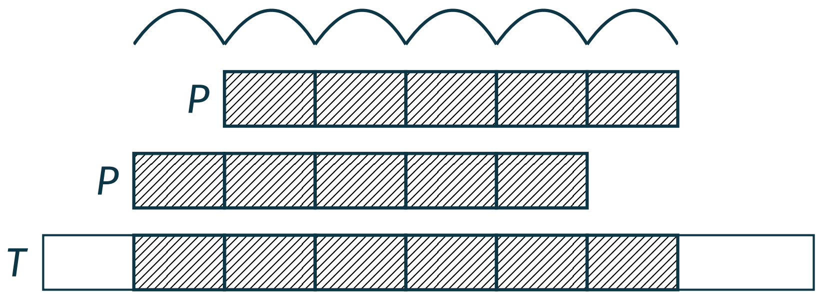
- The pattern P has at most one occurrence in T.
- The pattern P is periodic.



The Structure of Pattern Matching

Fact [folklore] Given a pattern P of length m and a text T of length $n \leq {}^3\!/_2\, m$ at least one of the following holds:

- The pattern P has at most one occurrence in T.
- The pattern P is periodic.



The fragment of T spanned by P's occurrences is periodic as well.

The Structure of Pattern Matching under Edit Distance

The Structure of Pattern Matching under Edit Distance

Theorem [CKW; FOCS'20] Given a pattern P of length m and a text T of length $n \leq {}^{3}\!/_{2}\, m$, and a threshold $k \leq m$ at least one of the following holds:

The Structure of Pattern Matching under Edit Distance

Theorem [CKW; FOCS'20] Given a pattern P of length m and a text T of length $n \leq {}^3\!/_2\, m$, and a threshold $k \leq m$ at least one of the following holds:

- The pattern P has $\mathcal{O}(k^2)$ k-error occurrences in T.

The Structure of Pattern Matching under Edit Distance

Theorem [CKW; FOCS'20] Given a pattern P of length m and a text T of length $n \leq 3/2\, m$, and a threshold $k \leq m$ at least one of the following holds:

- The pattern P has $\mathcal{O}(k^2)$ k-error occurrences in T.
- The pattern is almost periodic: at edit distance $< 2k$ from a string with period $\mathcal{O}(m/k)$.

The Structure of Pattern Matching under Edit Distance

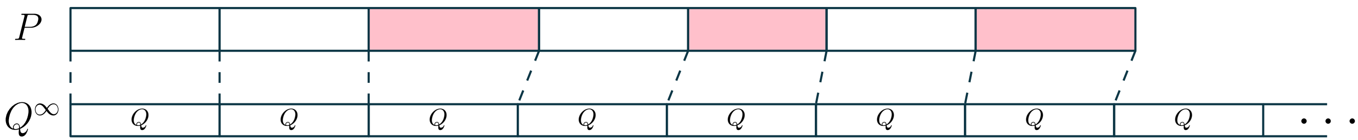
Theorem [CKW; FOCS'20] Given a pattern P of length m and a text T of length $n \leq {}^{3}\!/_{2}\, m$, and a threshold $k \leq m$ at least one of the following holds:

- The pattern P has $\mathcal{O}(k^2)$ k-error occurrences in T.
- The pattern is almost periodic: at edit distance $< 2k$ from a string with period $\mathcal{O}(m/k)$. **This is the bottleneck.**

The Structure of Pattern Matching under Edit Distance

Theorem [CKW; FOCS'20] Given a pattern P of length m and a text T of length $n \leq 3/2\, m$, and a threshold $k \leq m$ at least one of the following holds:

- The pattern P has $\mathcal{O}(k^2)$ k-error occurrences in T.
- The pattern is almost periodic: at edit distance $< 2k$ from a string with period $\mathcal{O}(m/k)$. **This is the bottleneck.**

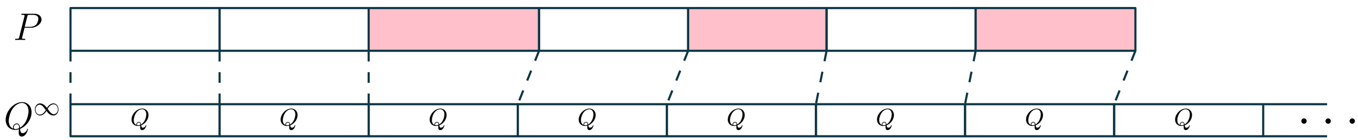


Q will denote a primitive string; it does not match any of its rotations.

The Structure of Pattern Matching under Edit Distance

Theorem [CKW; FOCS'20] Given a pattern P of length m and a text T of length $n \leq 3/2\, m$, and a threshold $k \leq m$ at least one of the following holds:

- The pattern P has $\mathcal{O}(k^2)$ k-error occurrences in T.
- The pattern is almost periodic: at edit distance $< 2k$ from a string with period $\mathcal{O}(m/k)$. **This is the bottleneck.**



Q will denote a primitive string; it does not match any of its rotations.

We call this a tile decomposition of P with respect to Q.

The PILLAR Model and the Reduction of [CKW'20]

The PILLAR Model and the Reduction of [CKW'20]

In the PILLAR model [CKW'20], algorithms rely on primitive operations.

The PILLAR Model and the Reduction of [CKW'20]

In the PILLAR model [CKW'20], algorithms rely on primitive operations.

For any setting, e.g., when the strings are given in compressed form, an efficient implementation of the primitive operations yields a fast algorithm.

The PILLAR Model and the Reduction of [CKW'20]

In the PILLAR model [CKW'20], algorithms rely on primitive operations.

For any setting, e.g., when the strings are given in compressed form, an efficient implementation of the primitive operations yields a fast algorithm.

Standard setting: The primitive operations take $\mathcal{O}(1)$ time after an $\mathcal{O}(n)$-time preprocessing.

The PILLAR Model and the Reduction of [CKW'20]

In the PILLAR model [CKW'20], algorithms rely on primitive operations.

For any setting, e.g., when the strings are given in compressed form, an efficient implementation of the primitive operations yields a fast algorithm.

Standard setting: The primitive operations take $\mathcal{O}(1)$ time after an $\mathcal{O}(n)$-time preprocessing.

$\mathcal{O}(k^4 \cdot n/m)$ PILLAR-time algorithm [CKW'20] matches [Cole, Hariharan; SICOMP 2002] for the standard setting.

The PILLAR Model and the Reduction of [CKW'20]

In the PILLAR model [CKW'20], algorithms rely on primitive operations.

For any setting, e.g., when the strings are given in compressed form, an efficient implementation of the primitive operations yields a fast algorithm.

Standard setting: The primitive operations take $\mathcal{O}(1)$ time after an $\mathcal{O}(n)$-time preprocessing.

$\mathcal{O}(k^4 \cdot n/m)$ PILLAR-time algorithm [CKW'20] matches [Cole, Hariharan; SICOMP 2002] for the standard setting.

Reduction [CKW'20]: An algorithm that solves the almost periodic case in $\tilde{\mathcal{O}}(k^a \cdot n/m)$ PILLAR-time, for $a \geq 3$, implies an algorithm that solves the general case in $\tilde{\mathcal{O}}(k^a \cdot n/m)$ PILLAR-time.

Dynamic Puzzle Matching

Dynamic Puzzle Matching

Input: An integer k and a family $\mathcal{F}$ of strings containing a distinguished primitive string Q with $\sum_{F \in \mathcal{F}} \delta_E(F, Q) = \mathcal{O}(k)$.

Dynamic Puzzle Matching

Input: An integer k and a family $\mathcal{F}$ of strings containing a distinguished primitive string Q with $\sum_{F \in \mathcal{F}} \delta_E(F, Q) = \mathcal{O}(k)$.

Maintain: A sequence $\mathcal{I} = (U_1, V_1) \cdots (U_z, V_z)$ of pairs from $\mathcal{F}^2$.

Dynamic Puzzle Matching

Input: An integer k and a family $\mathcal{F}$ of strings containing a distinguished primitive string Q with $\sum_{F\in\mathcal{F}} \delta_E(F, Q) = \mathcal{O}(k)$.

Maintain: A sequence $\mathcal{I} = (U_1, V_1) \cdots (U_z, V_z)$ of pairs from $\mathcal{F}^2$.

Updates: Insertions and deletions of pairs in $\mathcal{I}$.

Dynamic Puzzle Matching

Input: An integer k and a family $\mathcal{F}$ of strings containing a distinguished primitive string Q with $\sum_{F\in\mathcal{F}} \delta_E(F, Q) = \mathcal{O}(k)$.

Maintain: A sequence $\mathcal{I} = (U_1, V_1) \cdots (U_z, V_z)$ of pairs from $\mathcal{F}^2$.

Updates: Insertions and deletions of pairs in $\mathcal{I}$.

Queries: Compute the k-error occurrences of $U_1 \cdots U_z$ in $V_1 \cdots V_z$.

Dynamic Puzzle Matching

Input: An integer k and a family $\mathcal{F}$ of strings containing a distinguished primitive string Q with $\sum_{F\in\mathcal{F}} \delta_E(F, Q) = \mathcal{O}(k)$.

Maintain: A sequence $\mathcal{I} = (U_1, V_1) \cdots (U_z, V_z)$ of pairs from $\mathcal{F}^2$.

Updates: Insertions and deletions of pairs in $\mathcal{I}$.

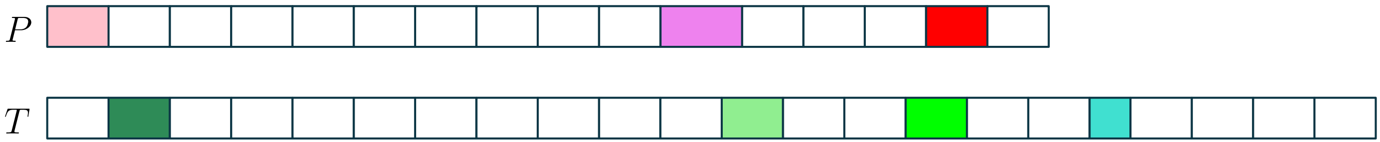
Queries: Compute the k-error occurrences of $U_1 \cdots U_z$ in $V_1 \cdots V_z$.

After $\tilde{\mathcal{O}}(k^3)$-time preprocessing, updates and queries take $\tilde{\mathcal{O}}(k)$ time.

Using Dynamic Puzzle Matching

Using Dynamic Puzzle Matching

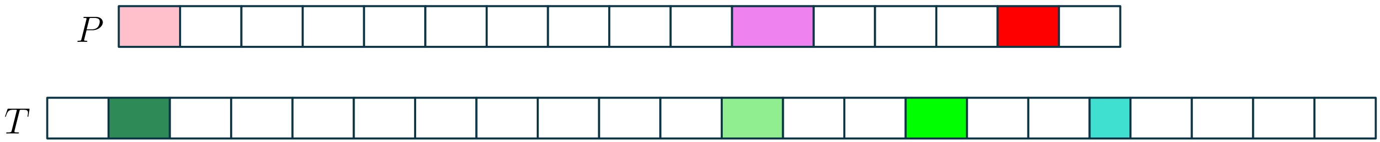
Think of: $k = 4$ and $|Q| \approx \sqrt{m}$.



Each string has $\mathcal{O}(k)$ special tiles.

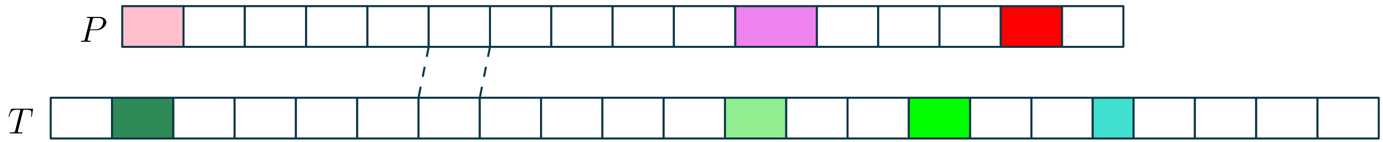
Using Dynamic Puzzle Matching

Think of: $k = 4$ and $|Q| \approx \sqrt{m}$.



Using Dynamic Puzzle Matching

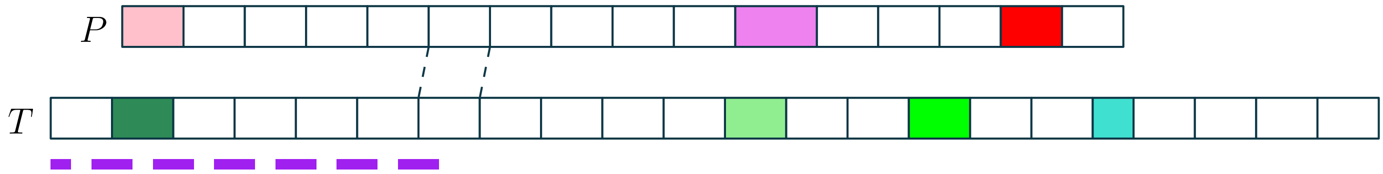
Think of: $k = 4$ and $|Q| \approx \sqrt{m}$.



$> k$ copies of Q in $P \implies \geq 1$ must be matched exactly

Using Dynamic Puzzle Matching

Think of: $k = 4$ and $|Q| \approx \sqrt{m}$.

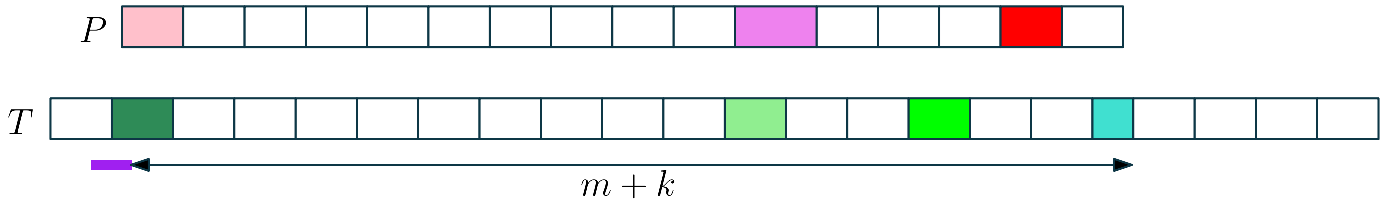


$> k$ copies of Q in $P \implies \geq 1$ must be matched exactly

Starting positions of k-error occs in T are within $\mathcal{O}(k)$ from endpoints of tiles.

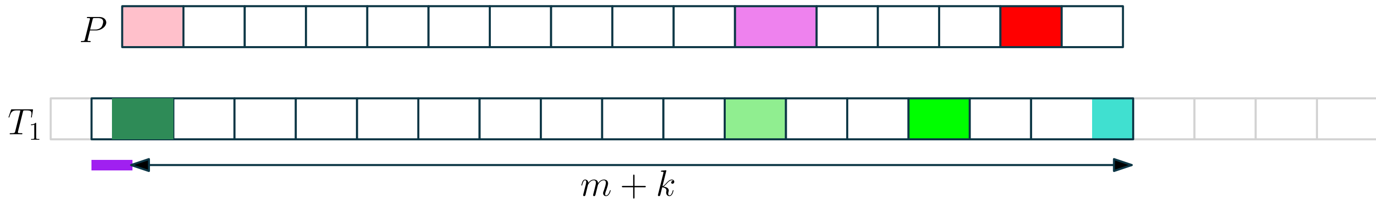
Using Dynamic Puzzle Matching

Think of: $k = 4$ and $|Q| \approx \sqrt{m}$.



Using Dynamic Puzzle Matching

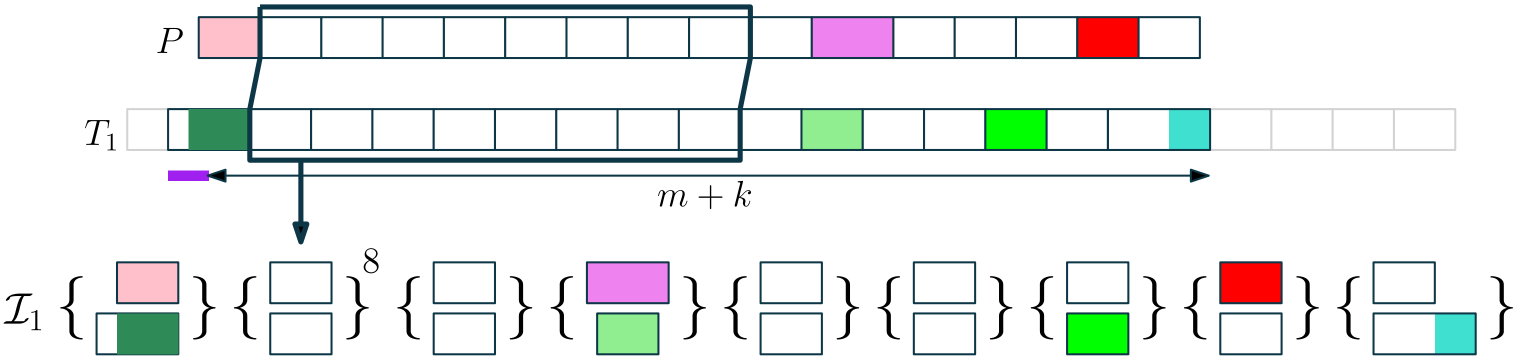
Think of: $k = 4$ and $|Q| \approx \sqrt{m}$.



$$|T_j| = m + \mathcal{O}(k)$$

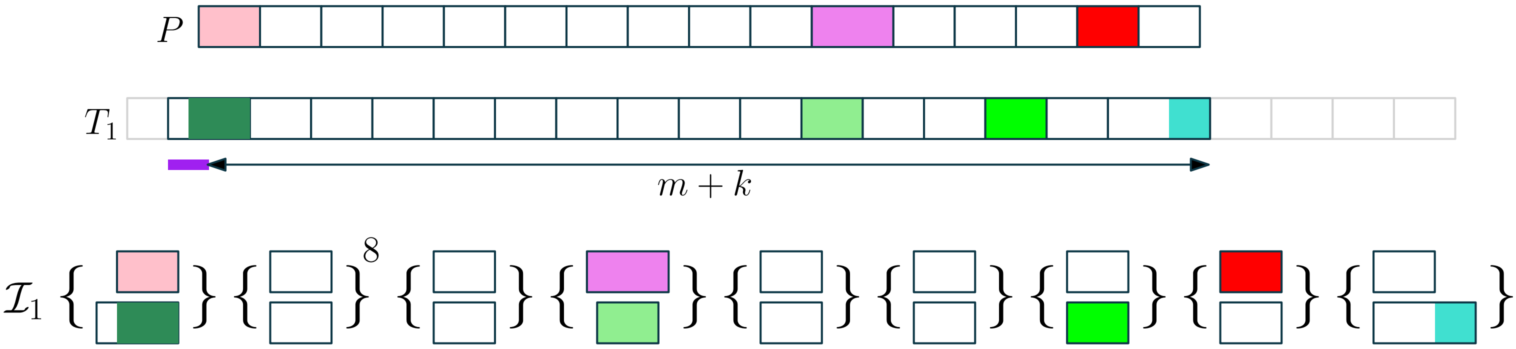
Using Dynamic Puzzle Matching

Think of: $k = 4$ and $|Q| \approx \sqrt{m}$.



Using Dynamic Puzzle Matching

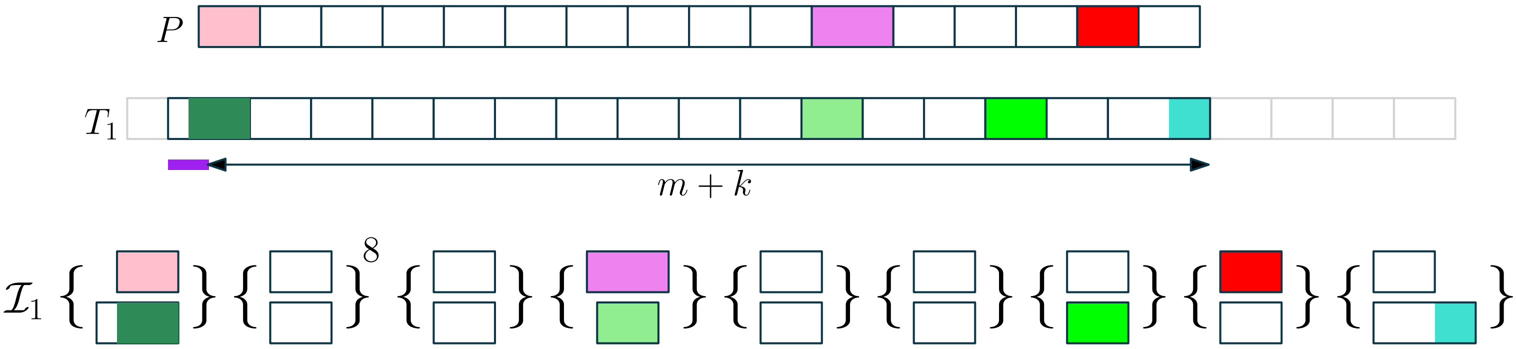
Think of: $k = 4$ and $|Q| \approx \sqrt{m}$.



Goal: Iterate over all $\mathcal{I}_j$'s in a DPM instance.

Using Dynamic Puzzle Matching

Think of: $k = 4$ and $|Q| \approx \sqrt{m}$.

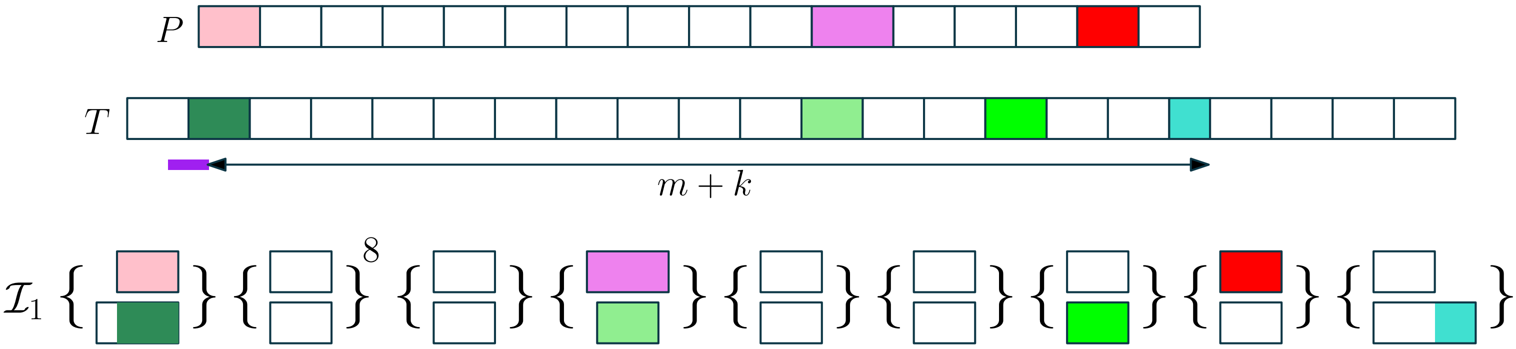


Goal: Iterate over all $\mathcal{I}_j$'s in a DPM instance.

(The leading and trailing pairs are treated separately.)

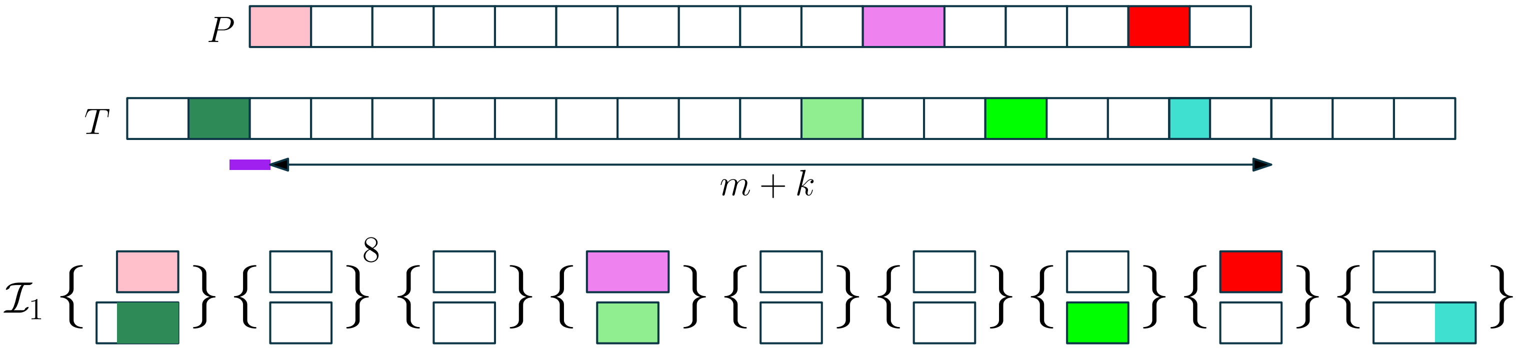
Using Dynamic Puzzle Matching

Think of: $k = 4$ and $|Q| \approx \sqrt{m}$.



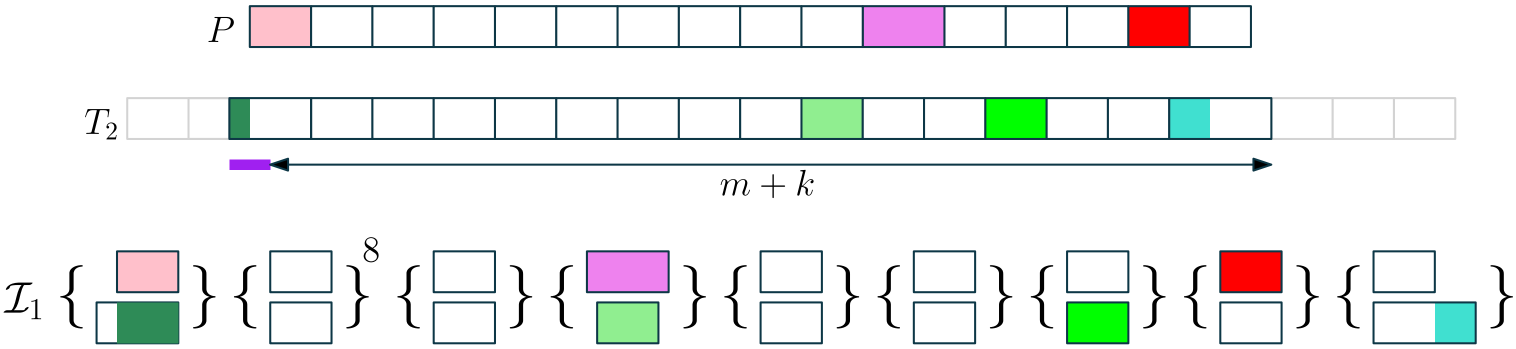
Using Dynamic Puzzle Matching

Think of: $k = 4$ and $|Q| \approx \sqrt{m}$.



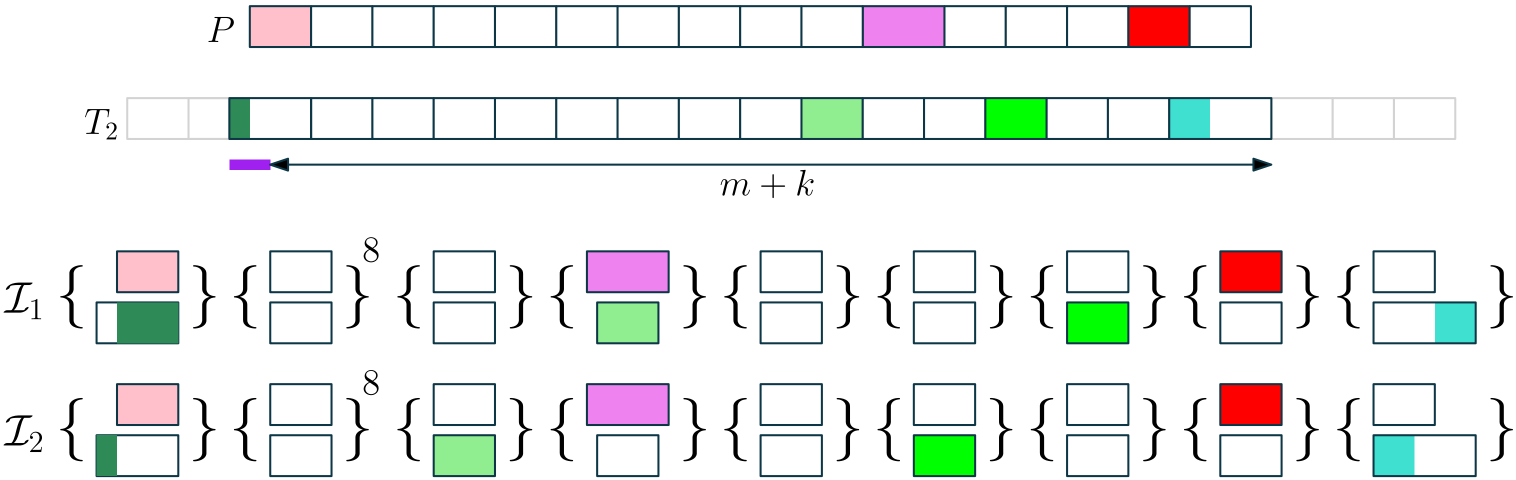
Using Dynamic Puzzle Matching

Think of: $k = 4$ and $|Q| \approx \sqrt{m}$.



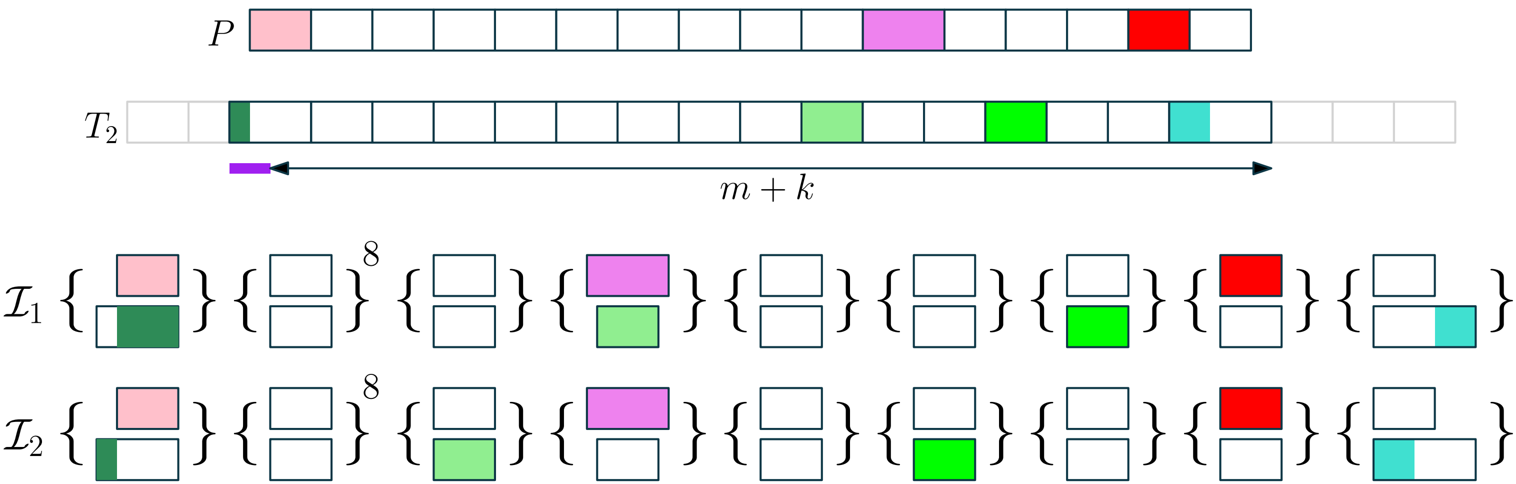
Using Dynamic Puzzle Matching

Think of: $k = 4$ and $|Q| \approx \sqrt{m}$.



Using Dynamic Puzzle Matching

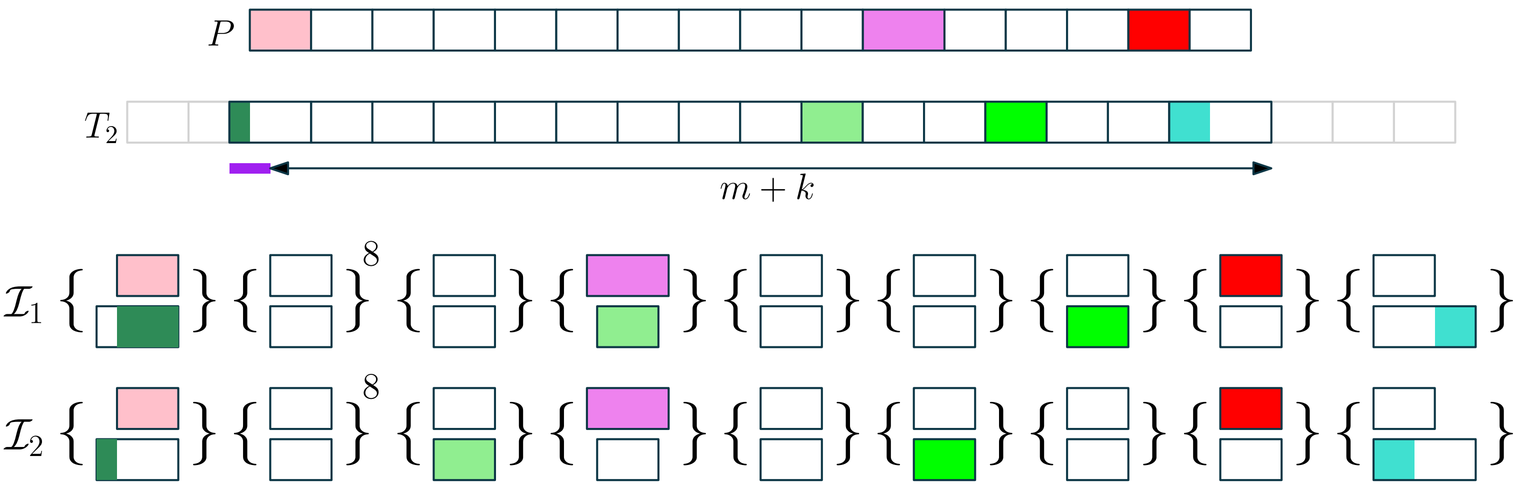
Think of: $k = 4$ and $|Q| \approx \sqrt{m}$.



We only need to update $\mathcal{O}(k)$ pairs; there has to be a pair $\neq (Q, Q)$ involved!

Using Dynamic Puzzle Matching

Think of: $k = 4$ and $|Q| \approx \sqrt{m}$.

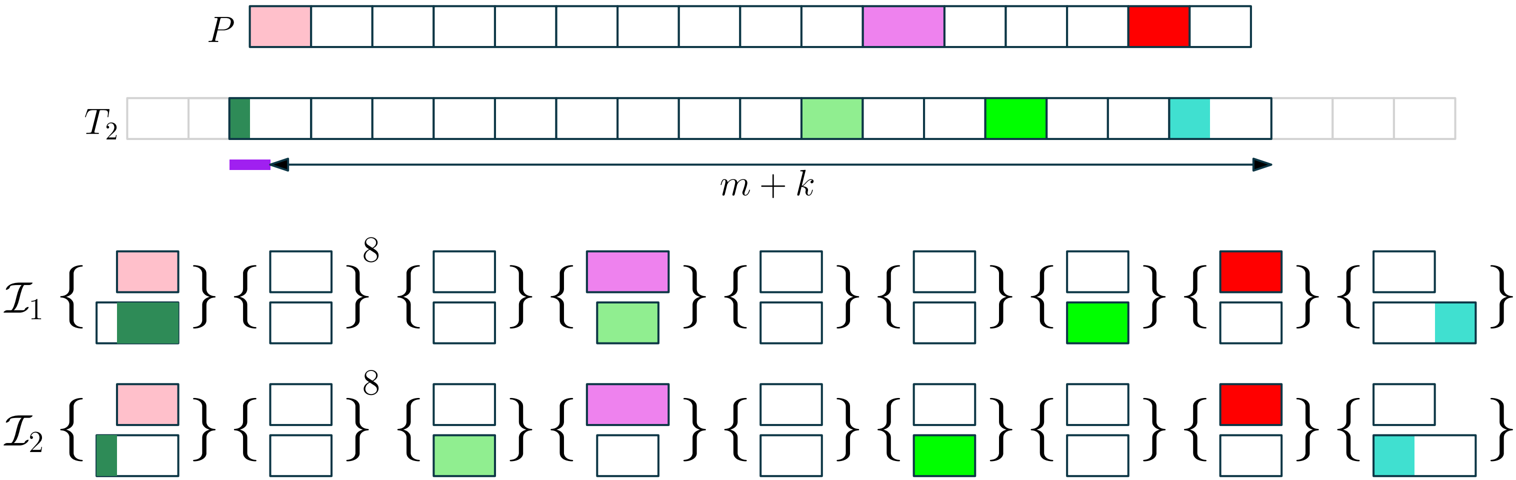


We only need to update $\mathcal{O}(k)$ pairs; there has to be a pair $\neq (Q, Q)$ involved!

Over the $\Theta(\sqrt{m})$ shifts of P, we need $\mathcal{O}(\sqrt{m} \cdot k)$ DPM-updates.

Using Dynamic Puzzle Matching

Think of: $k = 4$ and $|Q| \approx \sqrt{m}$.



We only need to update $\mathcal{O}(k)$ pairs; there has to be a pair $\neq (Q, Q)$ involved!

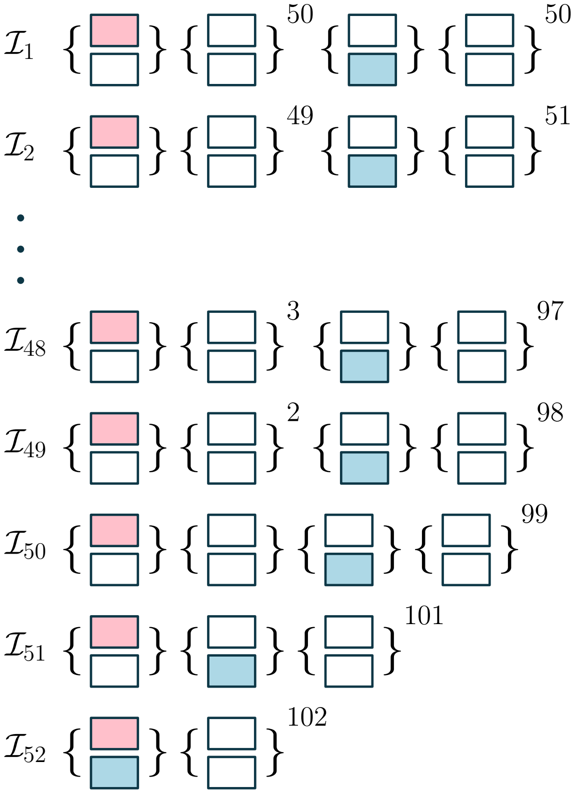
Over the $\Theta(\sqrt{m})$ shifts of P, we need $\mathcal{O}(\sqrt{m} \cdot k)$ DPM-updates.

Yields $\tilde{\mathcal{O}}(k^3 + \sqrt{m} \cdot k^2)$.

$\mathcal{O}(k^3)$ DPM-updates via Primitivity

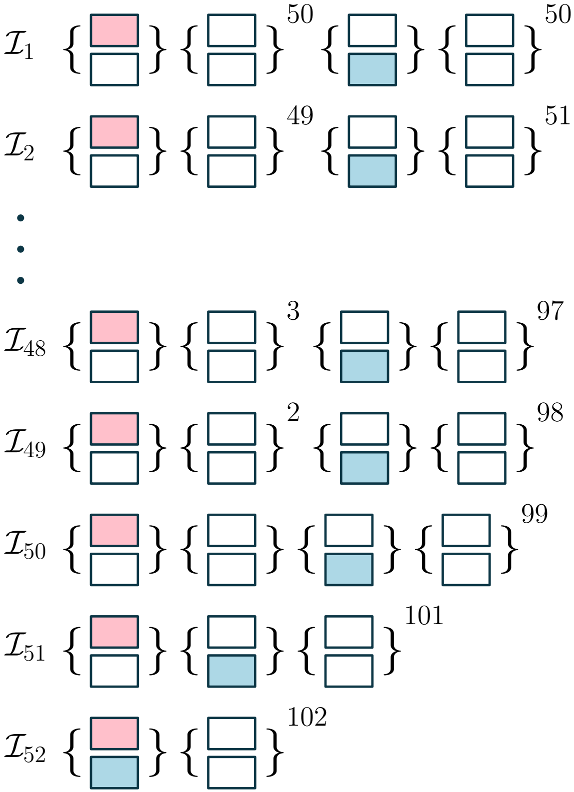
$\mathcal{O}(k^3)$ DPM-updates via Primitivity

$k = 2$



$\mathcal{O}(k^3)$ DPM-updates via Primitivity

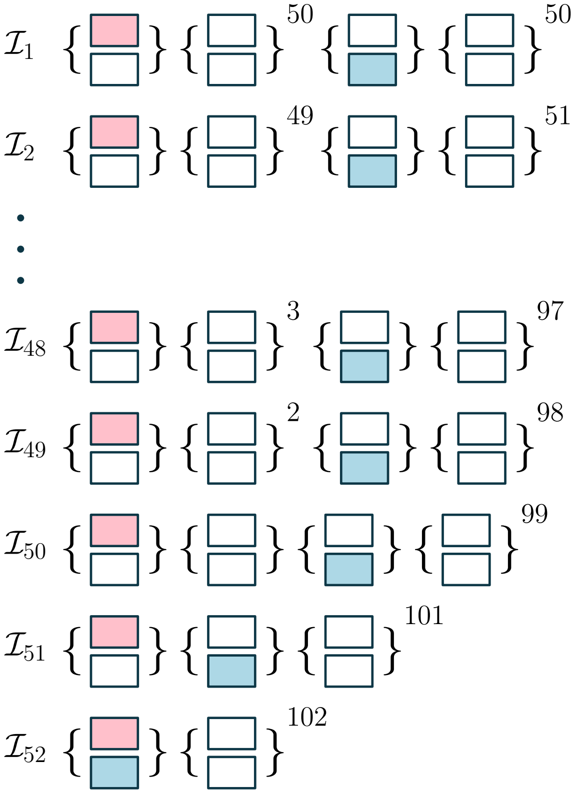
$k = 2$



For a plain run $(Q, Q)^y$, at least $y - k$ copies of Q will be matched exactly in a k-error occurrence.

$\mathcal{O}(k^3)$ DPM-updates via Primitivity

$k = 2$

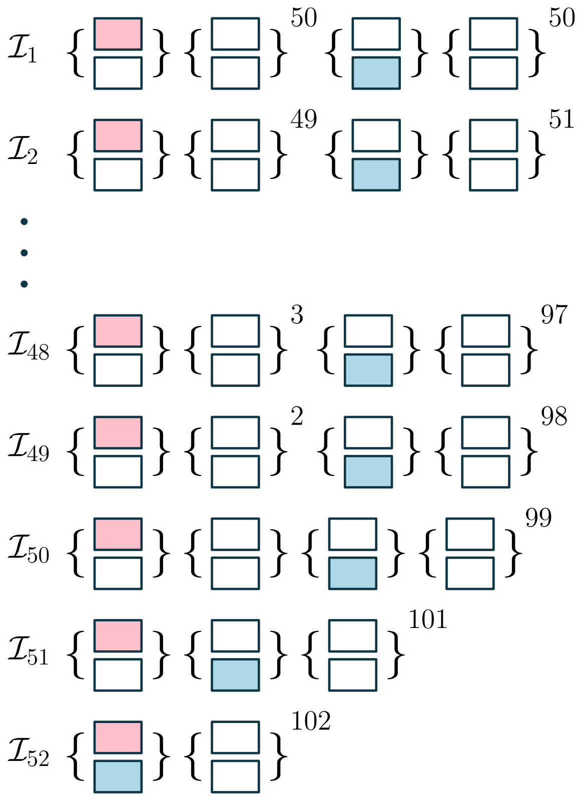


For a plain run $(Q, Q)^y$, at least $y - k$ copies of Q will be matched exactly in a k-error occurrence.

Cap exponents of plain runs at $k + 1$.

$\mathcal{O}(k^3)$ DPM-updates via Primitivity

$k = 2$



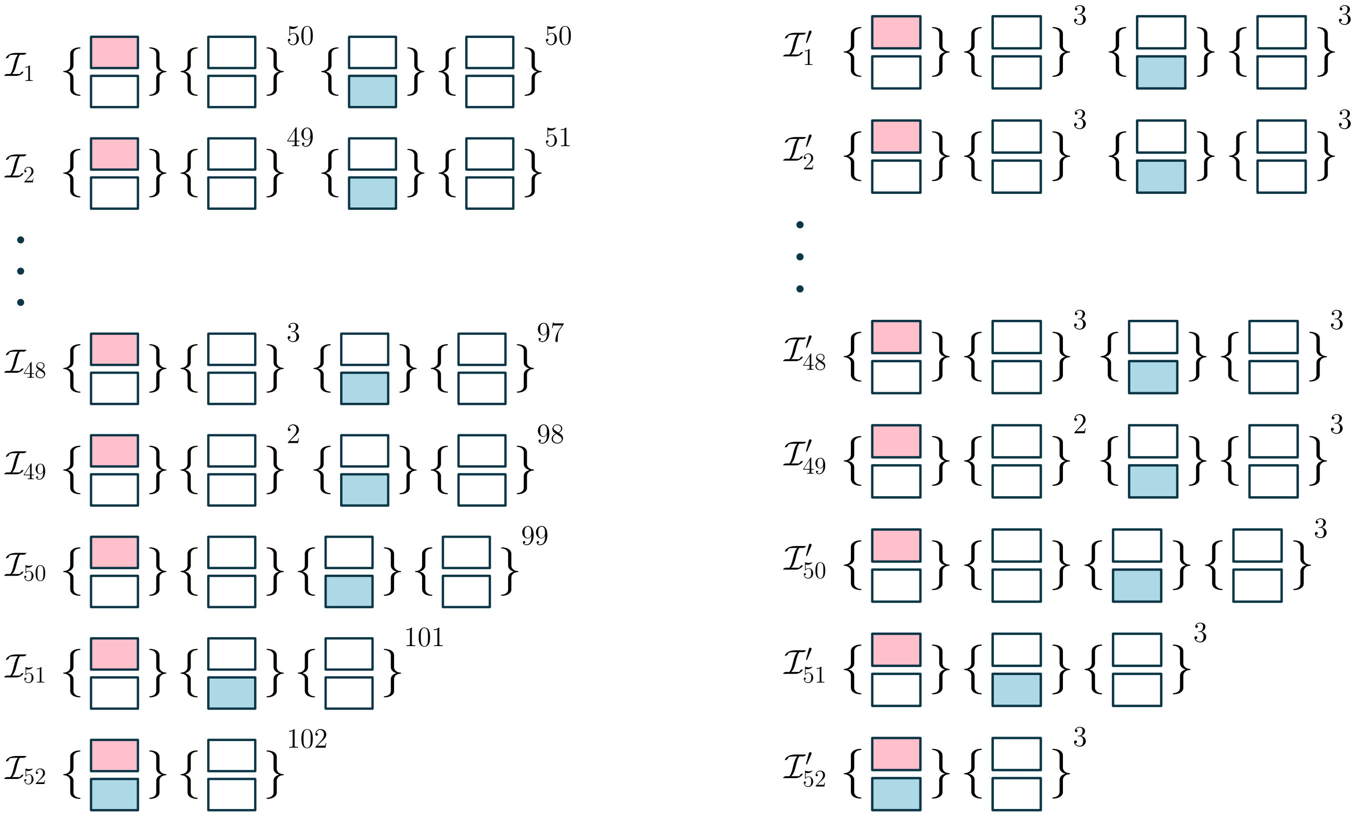
For a plain run $(Q, Q)^y$, at least $y - k$ copies of Q will be matched exactly in a k-error occurrence.

Cap exponents of plain runs at $k + 1$.

We do not lose or gain any k-error occs.

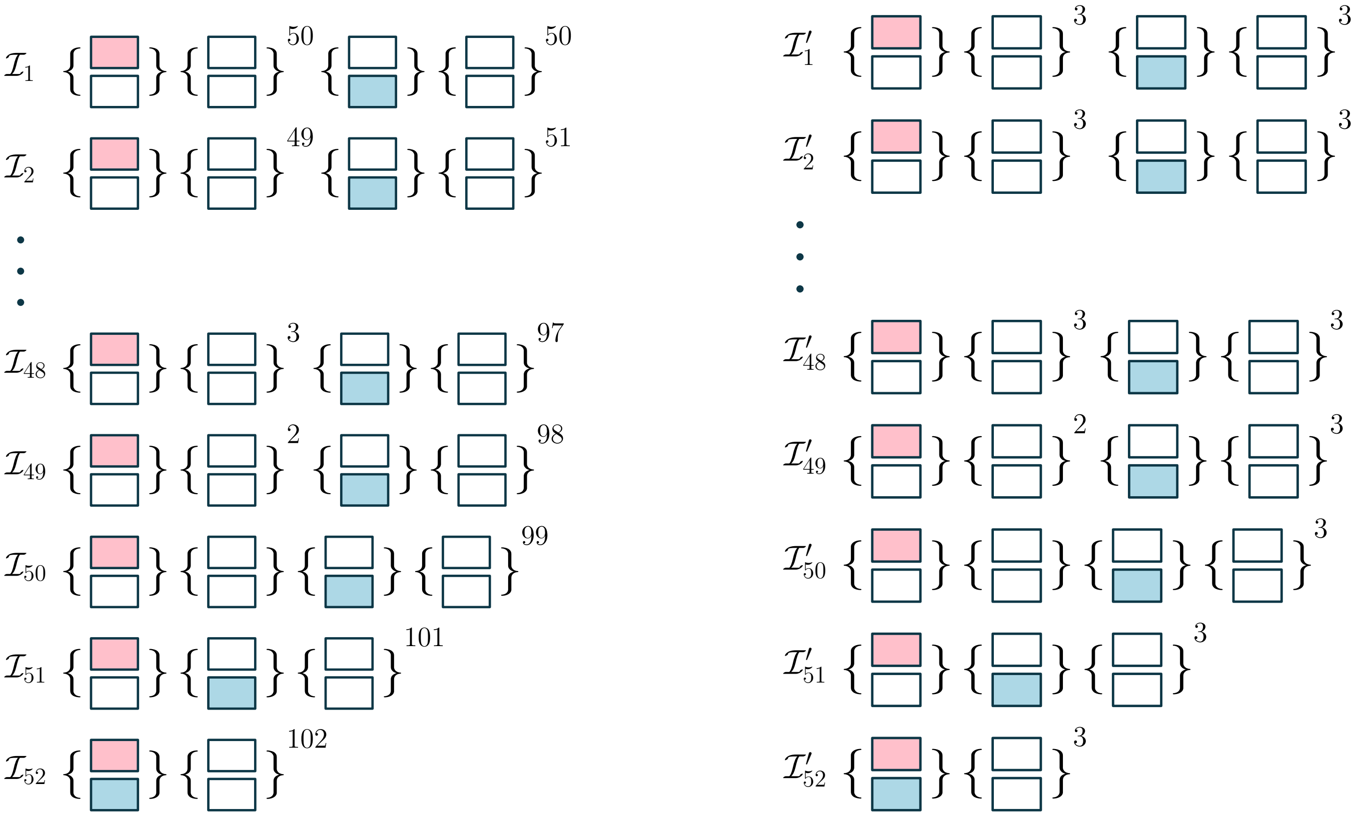
$\mathcal{O}(k^3)$ DPM-updates via Primitivity

$k = 2$



$\mathcal{O}(k^3)$ DPM-updates via Primitivity

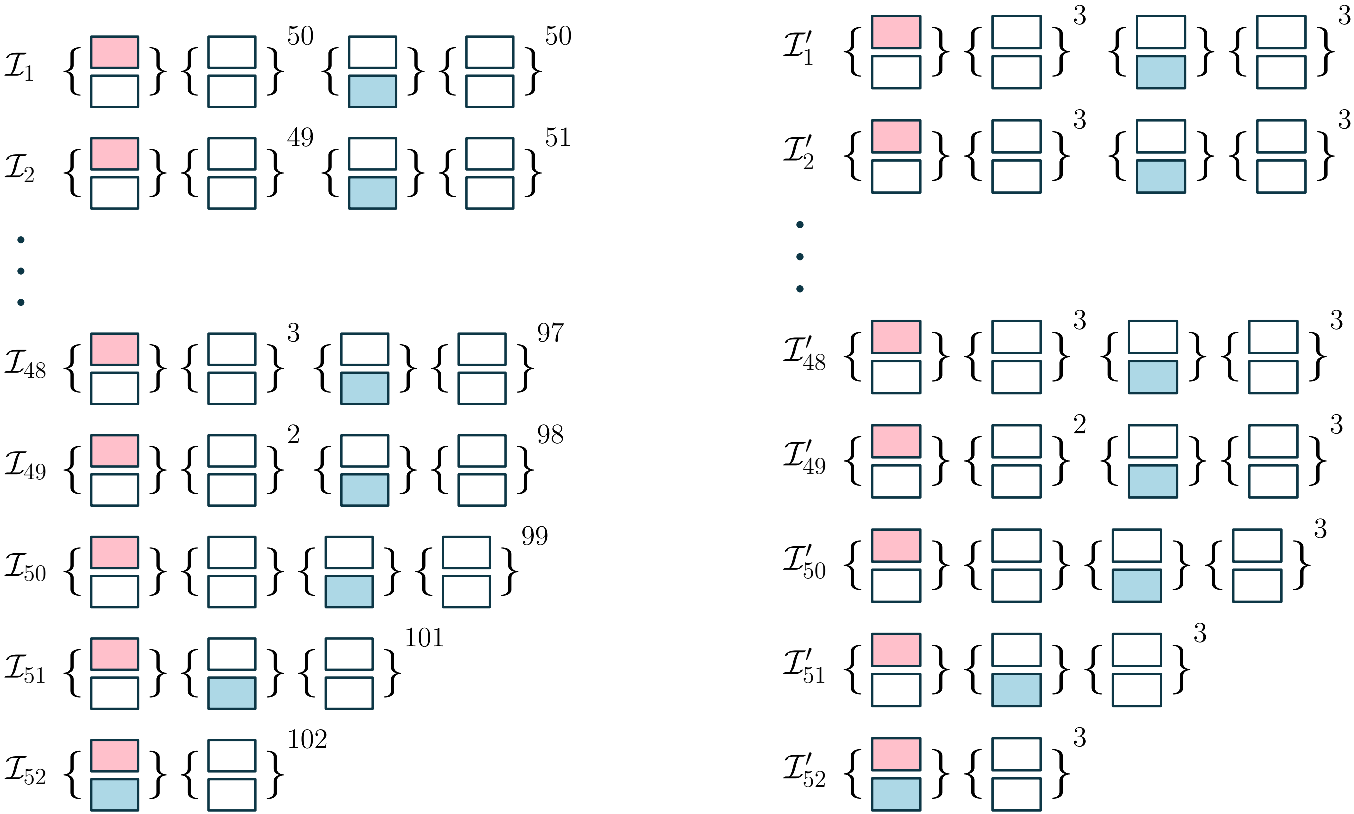
$k = 2$



The shown pair of special tiles implies $\mathcal{O}(k)$ DPM-updates.

$\mathcal{O}(k^3)$ DPM-updates via Primitivity

$k = 2$

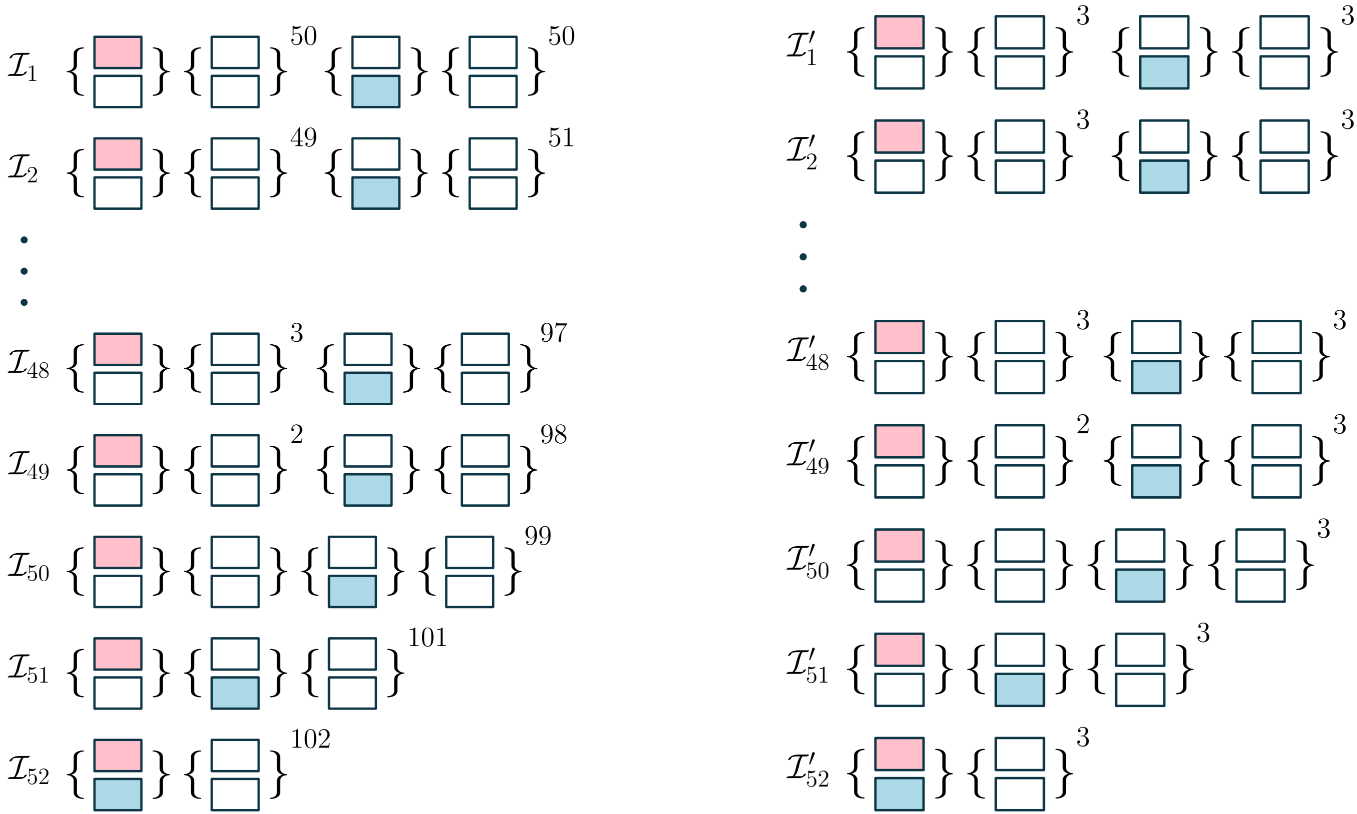


The shown pair of special tiles implies $\mathcal{O}(k)$ DPM-updates.

We have $\mathcal{O}(k^2)$ pairs of special tiles!

$\mathcal{O}(k^3)$ DPM-updates via Primitivity

$k = 2$



Alternative $\tilde{\mathcal{O}}(k^4)$-time algorithm!

Overview for $\mathcal{O}(k^{2.5})$ DPM-updates

Overview for $\mathcal{O}(k^{2.5})$ DPM-updates

Cap exponents of plain runs at $\sqrt{k}$.

Overview for $\mathcal{O}(k^{2.5})$ DPM-updates

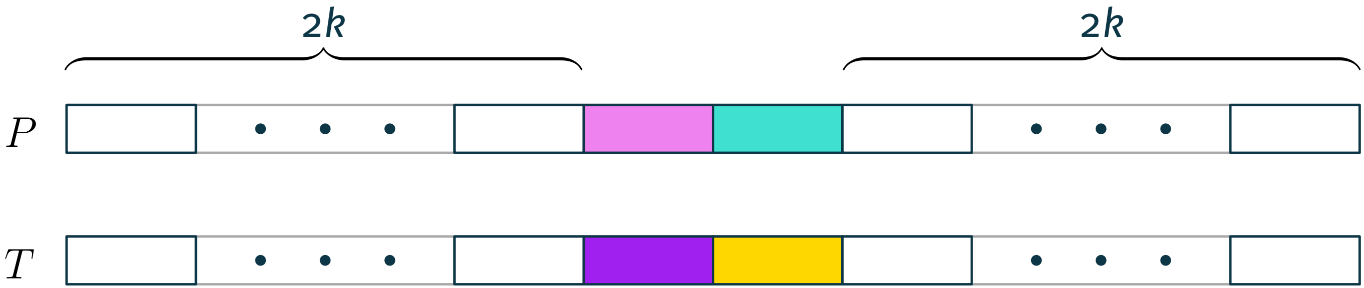
Cap exponents of plain runs at $\sqrt{k}$.

We may get false positives when we have $\geq \sqrt{k}$ edits in a run of (Q, Q).

Overview for $\mathcal{O}(k^{2.5})$ DPM-updates

Cap exponents of plain runs at $\sqrt{k}$.

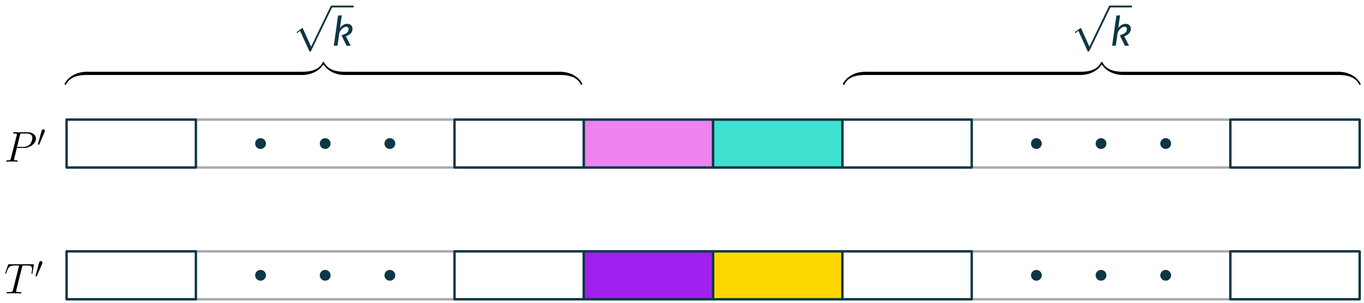
We may get false positives when we have $\geq \sqrt{k}$ edits in a run of (Q, Q).



Overview for $\mathcal{O}(k^{2.5})$ DPM-updates

Cap exponents of plain runs at $\sqrt{k}$.

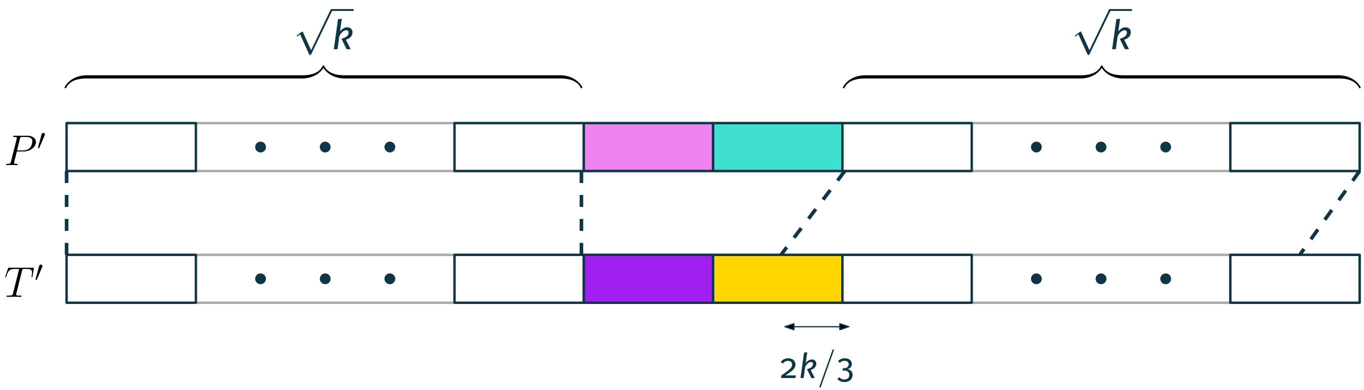
We may get false positives when we have $\geq \sqrt{k}$ edits in a run of (Q, Q).



Overview for $\mathcal{O}(k^{2.5})$ DPM-updates

Cap exponents of plain runs at $\sqrt{k}$.

We may get false positives when we have $\geq \sqrt{k}$ edits in a run of (Q, Q).



Cost: $0 + 0 + \sqrt{k} \cdot \delta_E(Q, rot^{2k/3}(Q))$.

Overview for $\mathcal{O}(k^{2.5})$ DPM-updates

Cap exponents of plain runs at $\sqrt{k}$.

We may get false positives when we have $\geq \sqrt{k}$ edits in a run of (Q, Q).

In this case, we must be saving $\geq \sqrt{k}$ by canceling out errors between P and Q^∞ with errors between T and Q^∞.

Overview for $\mathcal{O}(k^{2.5})$ DPM-updates

Cap exponents of plain runs at $\sqrt{k}$.

We may get false positives when we have $\geq \sqrt{k}$ edits in a run of (Q, Q).

In this case, we must be saving $\geq \sqrt{k}$ by canceling out errors between P and Q^∞ with errors between T and Q^∞.

We quantify potential savings using a marking scheme based on overlaps of special tiles and verify $\mathcal{O}(k^{2.5})$ positions with $\geq \sqrt{k}$ marks using known techniques.

Overview for $\mathcal{O}(k^{2.5})$ DPM-updates

Cap exponents of plain runs at $\sqrt{k}$.

We may get false positives when we have $\geq \sqrt{k}$ edits in a run of (Q, Q).

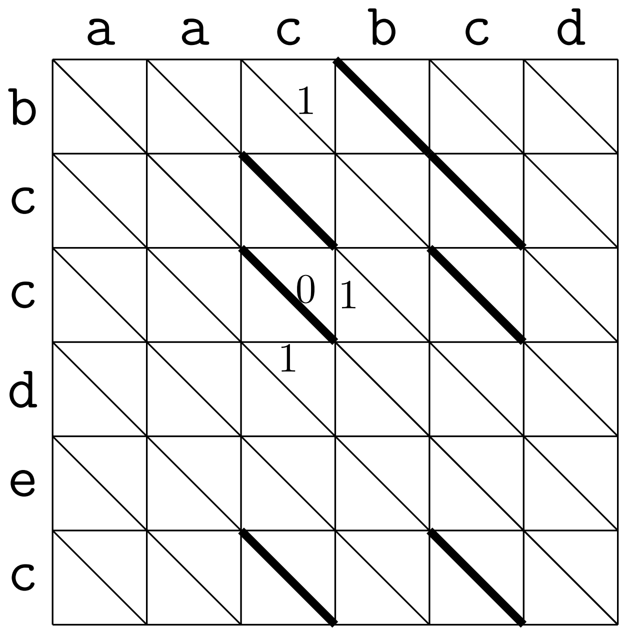
In this case, we must be saving $\geq \sqrt{k}$ by canceling out errors between P and Q^∞ with errors between T and Q^∞.

We quantify potential savings using a marking scheme based on overlaps of special tiles and verify $\mathcal{O}(k^{2.5})$ positions with $\geq \sqrt{k}$ marks using known techniques.

This yields $\mathcal{O}(k^{2.5})$ DPM-updates and hence $\tilde{\mathcal{O}}(k^{3.5})$ time overall.

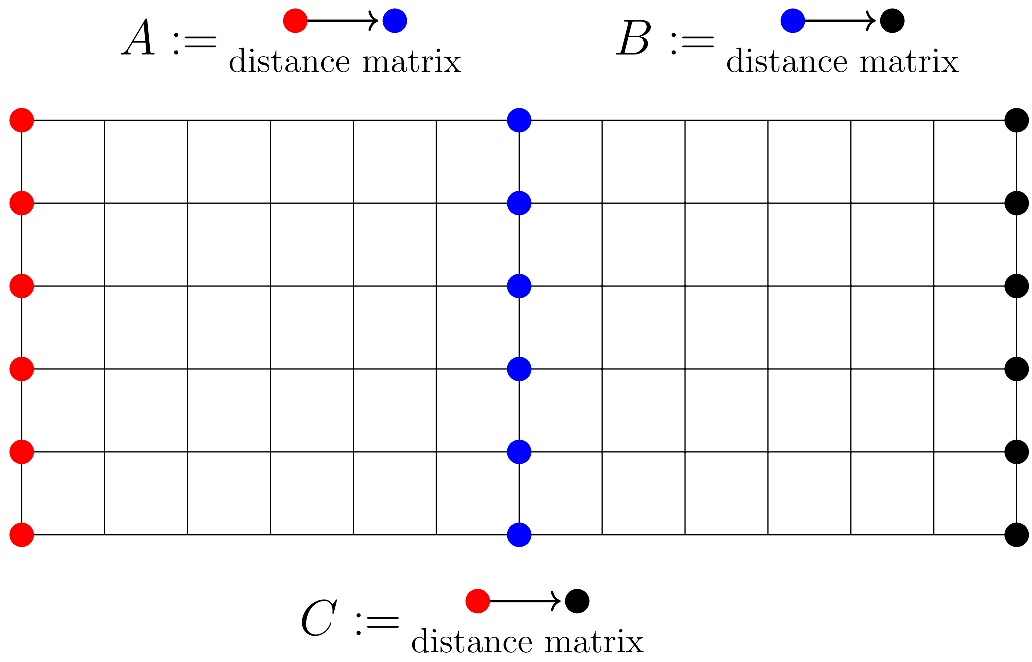
A Solution to DPM and a Grid View

A Solution to DPM and a Grid View

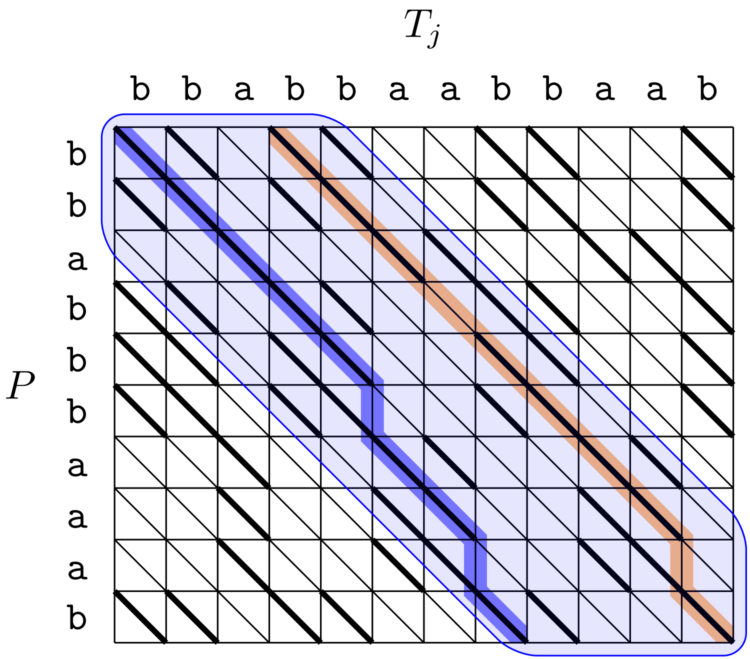


A Solution to DPM and a Grid View

Theorem [Tiskin; Algorithmica 2015] Matrix C can be computed from (small representations of) $n \times n$ matrices A and B in $\mathcal{O}(n \log n)$ time.



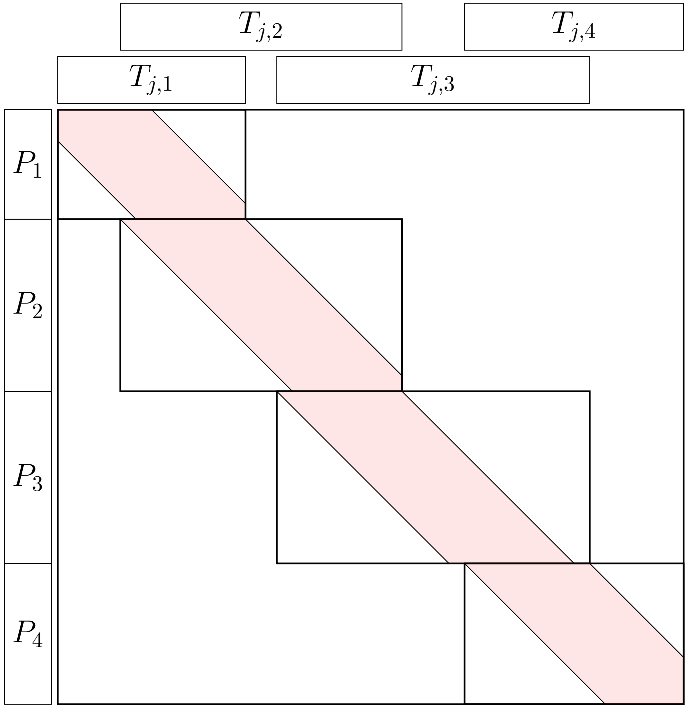
A Solution to DPM and a Grid View



$P = 10$, $T_j = 12$, $k = 2$.

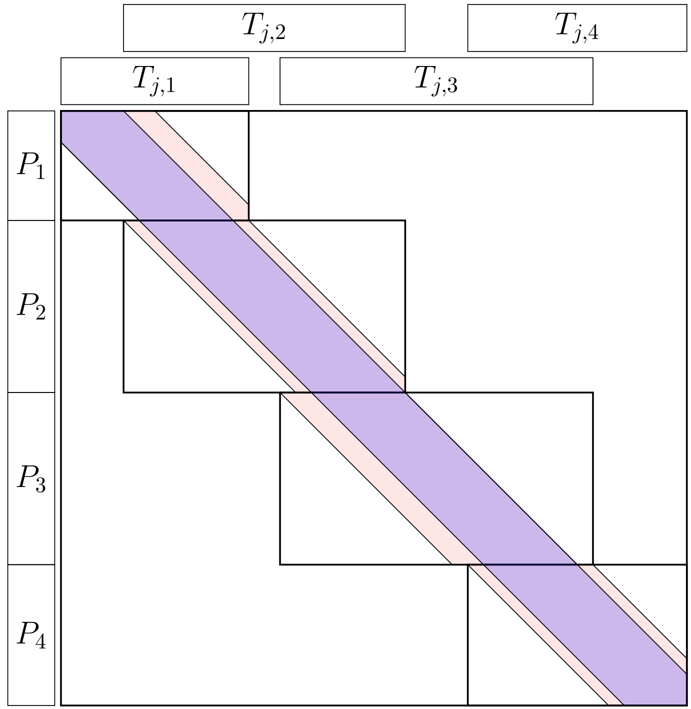
Only $|T_j| - |P| + 2k + 1 = \mathcal{O}(k)$ diagonals are relevant.

A Solution to DPM and a Grid View



Preprocessing: Build distance matrices for these small alignment grids.

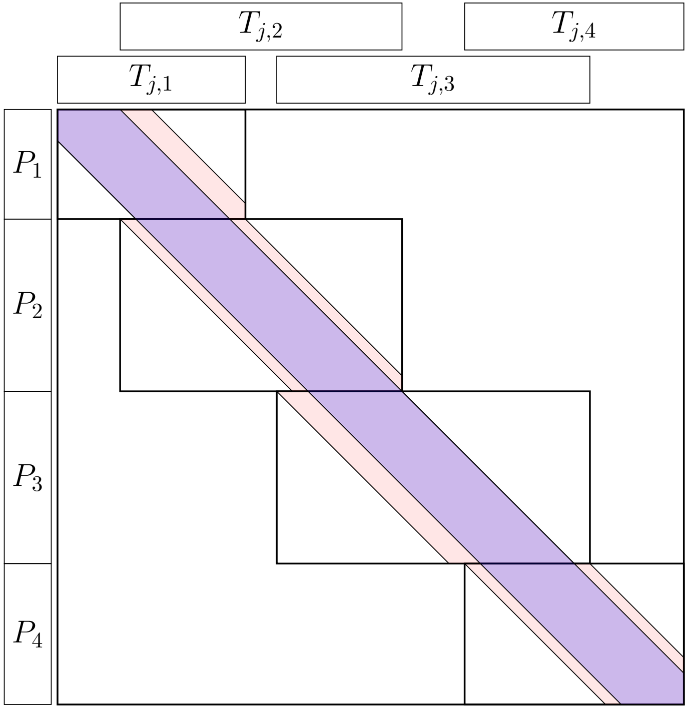
A Solution to DPM and a Grid View



Preprocessing: Build distance matrices for these small alignment grids.

Update: Maintain a balanced binary tree over them, stitching them together.

A Solution to DPM and a Grid View



Preprocessing: Build distance matrices for these small alignment grids.

Update: Maintain a balanced binary tree over them, stitching them together.

Each stitching operation takes $\tilde{O}(k)$ time.

Final Remarks and Open Problems

Final Remarks and Open Problems

What is the right exponent?

Cole and Hariharan's conjecture: $\mathcal{O}(n + k^3 \cdot n/m)$ *should be possible.*

Final Remarks and Open Problems

What is the right exponent?

Cole and Hariharan's conjecture: $\mathcal{O}(n + k^3 \cdot n/m)$ *should be possible.*

Is the decision version easier?

Final Remarks and Open Problems

What is the right exponent?

Cole and Hariharan's conjecture: $\mathcal{O}(n + k^3 \cdot n/m)$ *should be possible.*

Is the decision version easier?

What if we allow for some approximation by also reporting an arbitrary subset of the positions in $\mathrm{Occ}^E_{(1+\epsilon)k}(P, T) \setminus \mathrm{Occ}^E_k(P, T)$ for a small $\epsilon > 0$?

Final Remarks and Open Problems

What is the right exponent?

Cole and Hariharan's conjecture: $\mathcal{O}(n + k^3 \cdot n/m)$ *should be possible.*

Is the decision version easier?

What if we allow for some approximation by also reporting an arbitrary subset of the positions in $\mathrm{Occ}^E_{(1+\epsilon)k}(P, T) \setminus \mathrm{Occ}^E_k(P, T)$ for a small $\epsilon > 0$?

We report starting positions. How fast can we report substrings?

The End

Thank you for your attention!