

Fast algorithm for two-dimensional pattern matching with k mismatches

Jonas Ellert¹, Paweł Gawrychowski², Adam Górkiewicz³, and Tatiana Starikovskaya⁴

¹?

²?

³?

⁴?

Abstract

1 Introduction

We consider the one-dimensional all-substring Hamming distance problem (HD1D), where for a given text string T of length n and a string P of length m ($m < n$), we want to calculate the Hamming distance between P and every fragment T of length m .

We consider the two-dimensional all-substring Hamming distance problem (HD2D), where for a given 2D string T of size $n \times n$ and a string P of size $m \times m$ ($m < n$), we want to calculate the Hamming distance between P and every $m \times m$ fragment of T .

We also consider the bounded variants of HD1D and HD2D, where we are only required to calculate the distances which are not greater than k , for some parameter $k \in \mathbb{Z}^+$.

Theorem 1 (Main result). *Bounded HD2D can be solved in $\tilde{O}((m^2 + mk^{5/4})n^2/m^2)$ time.*

2 Preliminaries

For our purposes we will not use the standard definition of a two-dimensional string, where we associate it with a two-dimensional array of characters, and instead we will define it more broadly. Although we will occasionally use the array notation, we will do it exclusively for $n \times m$ strings. For any $n \in \mathbb{Z}^+$ we will denote $[n] = \{0, \dots, n-1\}$. We will only consider integer points or vectors and we will use these terms interchangeably. For a function f and a subset of its domain X we will write $f \upharpoonright_X$ to denote the restriction of f to X . Our results hold under word-RAM model of computation.

Definition 1 (Two-dimensional string). We define a **string** S as a partial function $\mathbb{Z}^2 \rightarrow \Sigma$ which maps some arbitrary set of integer points, denoted as $\text{dom}(S)$, to characters. For simplicity we will write $u \in S$ to denote that $u \in \text{dom}(S)$. We say that a string S is **partitioned** into strings R_1, \dots, R_ℓ when the sets $\text{dom}(R_1), \dots, \text{dom}(R_\ell)$ partition $\text{dom}(S)$ and $R_i(u) = S(u)$ for all $u \in R_i$. We call a string S **monochromatic** when $S(u) = \sigma$ for every $u \in S$ for some $\sigma \in \Sigma$ and we will write $C(S)$ to denote the value σ . We say that S is $n \times m$ for some $n, m \in \mathbb{Z}^+$ when $\text{dom}(S) = [n] \times [m]$. Physically we represent a string as a list of point-character pairs.

Definition 2 (Shifting). For a set of points $V \subseteq \mathbb{Z}^2$ and a vector $u \in \mathbb{Z}^2$, we denote $V + u$ as $\{v + u : v \in V\}$. For a string S and a vector $u \in \mathbb{Z}^2$ we denote $S + u$ as a string R such that

$\text{dom}(R) = \text{dom}(S) + u$ and $R(v) = S(v - u)$ for $v \in \text{dom}(R)$. Intuitively, we shift the set of points while maintaining their character values.

Definition 3 (Hamming distance). For a pair of strings S, R we define

$$\text{Ham}(S, R) = |\{u : u \in \text{dom}(S) \cap \text{dom}(R), S(u) \neq R(u)\}|,$$

which corresponds to the number of mismatches between S and R .

Under such notation, the HD2D problem is equivalent to calculating the (bounded or unbounded) values of $\text{Ham}(P + q, T)$ for all $q \in \mathbb{Z}^2$ such that $\text{dom}(P + q) \subseteq \text{dom}(T)$ (so for $q \in [n - m]^2$).

Definition 4 (Don't care symbol). We define the **don't care** symbol as a special character which matches with every character. We will denote it with $?$. Unless stated otherwise, we assume it is not allowed in Σ and in both HD1D and HD2D every character present in T and P matches only with itself.

Definition 5 (Vector operators). For any $u \in \mathbb{Z}^2$ we refer to its coordinates as $u.x, u.y$. For $u, v \in \mathbb{Z}^2$ we denote $u \cdot v = u.x \cdot v.x + u.y \cdot v.y$ and $u \times v = u.x \cdot v.y - u.y \cdot v.x$. Note that alternatively $u \cdot v = |u||v| \cos \alpha$ and $u \times v = |u||v| \sin \alpha$ where α is the angle between u and v .

Definition 6 (Quadrants). We define the four **quadrants** as

$$\begin{aligned} \mathcal{Q}_1 &= (0, +\infty) \times [0, +\infty), \\ \mathcal{Q}_2 &= (-\infty, 0] \times (0, +\infty), \\ \mathcal{Q}_3 &= (-\infty, 0] \times (-\infty, 0], \\ \mathcal{Q}_4 &= [0, +\infty) \times (-\infty, 0]. \end{aligned}$$

3 One-dimensional generalizations

In this section we explore some of the methods used for one-dimensional strings. Specifically, as our goal is to generalize the solution for pattern matching with k mismatches described in [2], we are especially interested in two-dimensional variants of the techniques that were used to solve the one-dimensional case.

Theorem 2 (Instanting). *Consider an algorithm \mathcal{A} which solves HD2D (bounded or unbounded), but only when $2|n$ and $n \leq \frac{3}{2}m$. If its running time is $\mathcal{T}(m)$, then the general case can be solved in $\mathcal{O}(\mathcal{T}(m)n^2/m^2)$.*

Proof. Let $r = \lfloor m/2 \rfloor$ and let $n' = r + m - 1$ or $r + m$ if $r + m - 1$ is odd. We see that the set $N = [n']^2$ satisfies the conditions for the text domain. For any vector $q \in [n - m]^2$ we can find a vector u such that $r|u.x, r|u.y$ and $q - u \in [r]^2$, so we have $\text{Ham}(P + q, T) = \text{Ham}(P + q - u, T_u)$ where $T_u = (T - u) \upharpoonright_N$. If $T - u$ is not defined for some $v \in N$, we can pad $T_u(v)$ with any character. We see that $\text{dom}(P + q - u) \subseteq N = \text{dom}(T_u)$. There are $\mathcal{O}(n^2/m^2)$ possible vectors u and we run \mathcal{A} for every pair of T_u and P . \square

Theorem 3 (Kangaroo jumps). *Consider an $n \times n$ string T , $m \times m$ string P and set of vectors Q such that $\text{dom}(P + q) \subseteq \text{dom}(T)$ for every $q \in Q$. There exists an algorithm which calculates $d_q = \text{Ham}(P + q, T)$ for every $q \in Q$ in total time $\tilde{\mathcal{O}}(n^2 + \sum_{q \in Q} d_q)$.*

Proof. For the sake of clarity, we will temporarily switch to the classical array notation for strings. Let T_0, \dots, T_{n-m} denote an array of two-dimensional strings (arrays) such that $T_k[0..n - 1, 0..m - 1] = T[0..n - 1, k..k + m - 1]$. For every row $P[0], \dots, P[m - 1]$ of P and every row

$T_k[0], \dots, T_k[n-1]$ of every T_k we assign an integer identifier so that $\text{Id}(P[i]) = \text{Id}(T_k[j]) \Leftrightarrow P[i] = T_k[j]$ by using the KMR algorithm ([reference]) in $\tilde{O}(n^2)$.

We use the approach described in [kangaroo reference]. There exists a data structure (suffix array) which for a given one-dimensional array S allows us to detect all mismatches between any given two of its subarrays of equal length. It can be built in $\tilde{O}(|S|)$ and the query time is $\tilde{O}(d+1)$ where d is the number of mismatches. We construct the suffix array for the concatenation of the following arrays:

- the rows $P[i]$ for every i ,
- the rows $T[i]$ for every i ,
- the array $\text{Id}(P[0]) \text{Id}(P[1]) \dots \text{Id}(P[m-1])$,
- the arrays $\text{Id}(T_k[0]) \text{Id}(T_k[1]) \dots \text{Id}(T_k[n-1])$ for every k ,

the total length of which is $\mathcal{O}(n^2)$. Let us consider a problem of detecting mismatches between P and some $T' = T[j \dots j+m-1, k \dots k+m-1]$. We can first find all row indices i for which $P[i] \neq T'[i]$ by finding all mismatches between $\text{Id}(P[0]) \dots \text{Id}(P[m-1])$ and $\text{Id}(T_k[j]) \dots \text{Id}(T_k[j+m-1])$, which we do with query to the data structure. For every such i we can then find all mismatches between $P[i]$ and $T'[i]$ by querying $P[i]$ and $T[i+j][k \dots k+m-1]$. If the distance between P and T' is d , the first query takes $\tilde{O}(d+1)$ operations and all subsequent queries take $\tilde{O}(d+1)$ operations in total. \square

Lemma 1. HD1D with don't care symbols can be solved in $\tilde{O}(n|\Sigma|)$ by running $|\Sigma|$ instances of FFT.

Lemma 2. There exists a $(1+\varepsilon)$ -approximate algorithm (introduced in [3]) which solves HD1D with don't care symbols in $\tilde{O}(n)$.

Theorem 4. HD2D with don't care symbols can be solved in $\tilde{O}(n^2|\Sigma|)$.

Proof. We will again use the array notation. We construct one-dimensional strings \bar{T} and \bar{P} by concatenating subsequent rows $T[0], \dots, T[n-1]$ of T and rows $P[0], \dots, P[m-1]$ of P padded with don't care symbols:

$$\begin{aligned}\bar{T} &= T[0] \ T[1] \ \dots \ T[n-1], \\ \bar{P} &= P[0] \ ?^{n-m} \ P[1] \ ?^{n-m} \ \dots \ ?^{n-m} \ P[m-1].\end{aligned}$$

We run the algorithm from Lemma 1. The distance between $T[i \dots i+m-1, j \dots j+m-1]$ and P is equal to the distance between $\bar{T}[in+j \dots in+j+nm-n+m-1]$ and \bar{P} . \square

Theorem 5. There exists a $(1+\varepsilon)$ -approximate algorithm which solves HD2D with don't care symbols in $\tilde{O}(n^2)$.

Proof. Identical to Theorem 4, but we use the algorithm from Lemma 2 instead of Lemma 1. \square

The same reduction as in Theorem 4 can be applied for every HD1D solution which allows don't care symbols. Unfortunately, the most effective known algorithms for bounded HD1D rely on periodicity ([1], [2]) and inherently do not allow don't care symbols, thus, they cannot be easily generalized.

Observation 1 (Don't care padding). Every HD2D solution which allows don't care symbols (eg. the algorithms from Theorem 4 and Theorem 5) can be extended to also calculate the Hamming distance for occurrences of P which are not entirely contained in T . It can be done by padding the text with don't care symbols and it does not change the complexity of the solution.

4 Proof of Theorem Theorem 1

We show an algorithm which works in time $\tilde{\mathcal{O}}(m^2 + mk^{5/4})$ assuming $2|n$ and $m < n \leq \frac{3}{2}m$. By Theorem 2, our main result follows.

We start by running the algorithm from Theorem 5 with $\varepsilon = 1$. We construct the set Q as the set of such vectors $q \in \mathbb{Z}^2$ for which the estimated value of $\text{Ham}(P + q, T)$ is at most $2k$. For every $q \in \{0, \dots, n - m\}^2 \setminus Q$ we say that $\text{Ham}(P + q, T)$ equals ∞ . The next step is to calculate the exact value of $\text{Ham}(P + q, T)$ for every $q \in Q$.

Let us consider the case when $|Q| \leq 2m + m^2/k$. We can run the algorithm from Theorem 3 and by the fact that $\text{Ham}(P + q, T) \leq 4k$ for every $q \in Q$, it will perform $\tilde{\mathcal{O}}(m^2 + mk)$ operations. We are left with the case when $|Q| > 2m + m^2/k$, in which we take advantage of the fact that some strings $P + q$ for $q \in Q$ must have a large overlap and small Hamming distance from each other, and thus P must be periodic.

4.1 Two-dimensional periodicity

In this section we introduce a range of new tools related to two-dimensional periodicity. We then select some special periods of the pattern and show how to decompose it into some regularly structured monochromatic strings.

Definition 7 (Periodicity). Consider any vector $\delta \in \mathbb{Z}^2$. We say that a string S has an ℓ -period δ when

$$\text{Ham}(S + \delta, S) \leq \ell.$$

Lemma 3. For every $u, v \in Q$, the vector $u - v$ is an $8k$ -period of P .

Proof. $\text{Ham}(P + u - v, P) = \text{Ham}(P + u, P + v) \leq \text{Ham}(P + u, T) + \text{Ham}(P + v, T) \leq 4k + 4k$. \square

Theorem 6. For a given $\ell \in \mathbb{Z}^+$ and a set of points $U \subseteq [\ell + 1]^2$ such that $|U| > 4\ell$ there exist $s, t, s', t' \in U$ such that the following conditions hold for $w = t - s$ and $w' = t' - s'$:

- $0 < |w||w'| \leq 22 \frac{\ell^2}{|U|}$,
- $|\sin \alpha| \geq \frac{1}{2}$ where α is the angle between w and w' ,
- $w, w', -w, -w'$ are all contained in different quadrants.

Such w, w' can be found in $\tilde{\mathcal{O}}(|U|)$ operations.

Proof. See Section 4.3. \square

We run the algorithm from Theorem 6 on the set Q (where $\ell = n - m \leq m/2$, thus $|Q| > 2m + m^2/k \geq 4\ell$). We obtain vectors $\varphi \in \mathcal{Q}_4$ and $\psi \in \mathcal{Q}_1$ which by Lemma 3 are $\mathcal{O}(k)$ -periods of P . We will refer to those vectors throughout the rest of the description and we define $p = \varphi \times \psi$. Note that because $|Q| > 2m + m^2/k$, we have $p \leq |\varphi||\psi| = \mathcal{O}(\min\{m, k\})$.

Definition 8 (Lattice congruency). We define $\mathcal{L} = \{s\varphi + t\psi : s, t \in \mathbb{Z}\}$. We say that two vectors $u, v \in \mathbb{Z}^2$ are **lattice-congruent** and denote $u \equiv v$ when $u - v \in \mathcal{L}$ [Galil citation].

Lemma 4. There exists a set of points $\Gamma \subseteq \mathbb{Z}^2$ such that $|\Gamma| = p$ and every point $u \in \mathbb{Z}^2$ is lattice-congruent to exactly one point $\gamma \in \Gamma$.

Proof. TODO \square

Definition 9 (Parquet). We call a non-empty set $U \subseteq \mathbb{Z}^2$ a **parquet** when there exist some values $x_0, x_1, y_0, y_1, \varphi_0, \varphi_1, \psi_0, \psi_1 \in \mathbb{Z}$, which we will call its **signature**, such that

$$U = [x_0, x_1) \times [y_0, y_1) \cap \{u : u \in \mathbb{Z}^2, \varphi \times u \in [\varphi_0, \varphi_1), \psi \times u \in [\psi_0, \psi_1)\}.$$

- a) If additionally $x_1 - x_0 \geq |\varphi.x| + |\psi.x|$ and $y_1 - y_0 \geq |\varphi.y| + |\psi.y|$, then U is a **spacious** parquet.
- b) If additionally $x_0, y_0 = -\infty$ and $x_1, y_1 = +\infty$, then U is a **simple** parquet.

Note that every simple parquet is spacious.

Definition 10 (Subparquet). We call a non-empty set $V \subseteq \mathbb{Z}^2$ a **subparquet** when there exists a parquet U and a point $\gamma \in \mathbb{Z}^2$ such that

$$V = \{u : u \in U, u \equiv \gamma\}.$$

We call V a spacious/simple subparquet when there exists U which is (correspondingly) a spacious/simple parquet. We say that V is lattice-congruent to some $v \in \mathbb{Z}^2$ (denoted as $V \equiv v$) when $v \equiv \gamma$. We similarly define lattice congruency between two subparquets.

Theorem 7 (Subparquet convolution). *For a given list of signatures of simple subparquets U_1, \dots, U_ℓ , list of signatures of subparquets V_1, \dots, V_ℓ and a set of vectors Q we can calculate*

$$\sum_{i=1}^{\ell} |(U_i + q) \cap V_i|$$

for every $q \in Q$ in total time $\tilde{O}(m^2 + \ell + |Q|)$ assuming that the subparquets only contain vectors of length $O(m)$.

Proof. See Section 4.4. □

Definition 11 (Parquet string). We call a string S a spacious/simple (sub-)parquet string when $\text{dom}(S)$ is a spacious/simple (sub-)parquet.

Theorem 8 (Periodic string decomposition). *A given spacious/simple parquet string R with $O(k)$ -periods φ and ψ can be partitioned in time $\tilde{O}(|\text{dom}(R)|)$ into $O(k)$ monochromatic spacious/simple subparquet strings, correspondingly.*

Proof. See Section 4.5. □

Since $|\varphi.x|, |\varphi.y|, |\psi.x|, |\psi.y| \leq n - m \leq m/2$, the $m \times m$ string P is a spacious parquet string and satisfies the assumptions of Theorem 8. We partition P into a set of strings \mathcal{V} . We then group the strings based on the single character they contain, specifically, we define $\mathcal{V}_\sigma = \{V : V \in \mathcal{V}, C(V) = \sigma\}$.

Theorem 9. *For a given set of monochromatic simple subparquet strings \mathcal{S} we can calculate*

$$\sum_{S \in \mathcal{S}} \text{Ham}(P + q, S)$$

for every $q \in Q$ in total time $\tilde{O}(m^2 + \sum_{S \in \mathcal{S}} |\mathcal{V}_{C(S)}|)$ assuming that $\text{dom}(S)$ for $S \in \mathcal{S}$ are some pairwise disjoint subsets of $\text{dom}(T)$.

Proof. Let $U = \bigcup_{S \in \mathcal{S}} \text{dom}(S)$. Observe that

$$\sum_{S \in \mathcal{S}} \text{Ham}(P + q, S) = |(P + q) \cap U| - \sum_{S \in \mathcal{S}} \sum_{V \in \mathcal{V}_{C(S)}} |\text{dom}(V + q) \cap \text{dom}(S)|.$$

We can calculate $|(P + q) \cap U|$ for every $q \in Q$ with a single instance of FFT (see Theorem 4) in time $\tilde{O}(m^2)$. To calculate

$$\sum_{S \in \mathcal{S}} \sum_{V \in \mathcal{V}_{C(S)}} |\text{dom}(V + q) \cap \text{dom}(S)|$$

we use the algorithm from Theorem 7 (with $\ell = \sum_{S \in \mathcal{S}} |\mathcal{V}_{C(S)}|$). □

4.2 Text decomposition

Because the text is not necessarily periodic, we unfortunately cannot use the same approach as for the pattern. In this section we show how to decompose the text using a similar but more nuanced approach.

Definition 12 (Active text). We define the **active text** $T_{\mathbf{a}}$ as the restriction of T to $\bigcup_{q \in Q} \text{dom}(P) + q$. For a point $u \in \mathbb{Z}^2$ we define its **border distance** as $\min \{ \|u - v\|_{\infty} : v \in \text{dom}(T) \setminus \text{dom}(T_{\mathbf{a}}) \}$, which we will denote as $\text{BD}(u)$. For a set of points $U \subseteq \mathbb{Z}^2$ we define $\text{BD}(U) = \max \{ \text{BD}(u) : u \in U \}$. Note that we consider the maximum distance, not minimum.

Observation 2. $\text{Ham}(P + q, T) = \text{Ham}(P + q, T_{\mathbf{a}})$ for every $q \in Q$.

Theorem 10 (Active text decomposition). For a given parameter $\ell \in \mathbb{N}$ the active text can be partitioned in time $\tilde{\mathcal{O}}(m^2)$ into a set of $\mathcal{O}(\min \{ m^2, \ell k \})$ monochromatic simple subparquet strings and a string F such that $\text{BD}(F) = \mathcal{O}(m/\ell)$.

Proof. See Section 4.6. □

Theorem 11. For a given string F such that $\text{dom}(F) \subseteq \text{dom}(T_{\mathbf{a}})$ we can calculate $\text{Ham}(P + q, F)$ for every $q \in Q$ in total time $\tilde{\mathcal{O}}(m^2 + mk^{1/2} \text{BD}(F))$.

Proof. See Section 4.8. □

We partition $T_{\mathbf{a}}$ using the algorithm from Theorem 10 with $\ell = mk^{-3/4}$ into a set of simple subparquet strings \mathcal{S} and a string F , where $|\mathcal{S}| = \mathcal{O}(mk^{1/4})$ and $\text{BD}(F) = \mathcal{O}(k^{3/4})$. For every $q \in Q$ we then have

$$\text{Ham}(P + q, T_{\mathbf{a}}) = \text{Ham}(P + q, F) + \sum_{S \in \mathcal{S}} \text{Ham}(P + q, S).$$

By Theorem 9, we can calculate $\sum_{S \in \mathcal{S}} \text{Ham}(P + q, S)$ for every $q \in Q$ in time $\tilde{\mathcal{O}}(m^2 + mk^{5/4})$, since $\sum_{S \in \mathcal{S}} |\mathcal{V}_{C(S)}| \leq |\mathcal{S}| |\mathcal{V}| = \mathcal{O}(mk^{5/4})$. By Theorem 11, we can calculate $\text{Ham}(P + q, F)$ for every $q \in Q$ in time $\tilde{\mathcal{O}}(m^2 + mk^{5/4})$, which ends the proof.

4.3 Proof of Theorem 6

First, we find any closest pair of vectors $s, t \in U$ by running the standard $\tilde{\mathcal{O}}(|U|)$ time algorithm and denote $w = t - s$. We define a partial order \leq_w where $v \leq_w u$ for some $u, v \in U$ when at least one condition holds:

- (a) $u = v$,
- (b) $u - v$ and w belong to the same quadrant,
- (c) $\alpha \in (-\frac{\pi}{6}, \frac{\pi}{6})$ where α is the angle between w and $u - v$.

We find the longest chain C and the longest antichain A using dynamic programming in $\tilde{\mathcal{O}}(|U|)$ operations. We then find any closest pair of vectors $s', t' \in A$ and denote $w' = t' - s'$. We have the following inequalities:

- (i) $|U| \leq |C| |A|$ (by Dilworth's theorem),
- (ii) $(|C| - 1)|w| \leq (1 + \sqrt{3})\ell$ (roughly by the fact that vectors in C must be increasing in a certain direction),
- (iii) $(|A| - 1)|w'| \leq 2\ell$ (by using a similar argument for vectors in A).

By considering the assumption $|U| > 4\ell$ it can be proven that $|w||w'| \leq 22 \frac{\ell^2}{|U|}$ and the other conditions also hold.

4.4 Proof of Theorem 7

Throughout this section we will denote $D = \{u : u \in \mathcal{L}, \varphi \times u \geq 0, \psi \times u \geq 0\}$, where \mathcal{L} is the set introduced in Definition 8.

Lemma 5. *Given a set of subparquets \mathcal{V} and a set of points Q , we can calculate*

$$\sum_{V \in \mathcal{V}} |(D + q) \cap V|$$

for every $q \in Q$ in total time $\tilde{\mathcal{O}}(n^2 + |Q| + |\mathcal{V}|)$ assuming that every $V \in \mathcal{V}$ consists of vectors of length $\mathcal{O}(n)$.

Proof. For every $u \in \mathbb{Z}^2$ let us define $\text{score}(u) = |\{V : V \in \mathcal{V}, u \in V\}|$. Observe that

$$\sum_{V \in \mathcal{V}} |(D + q) \cap V| = \sum_{u \in D+q} \text{score}(u).$$

We start by explicitly calculating the scores. We find the maximum length of a vector that some $V \in \mathcal{V}$ is defined for, which we denote ℓ . We construct the set $U \subseteq \mathbb{Z}^2$ of all vectors of length at most ℓ . By the assumption, we have $\ell = \mathcal{O}(n)$, and thus $|U| = \mathcal{O}(\ell^2) = \mathcal{O}(n^2)$. We observe that since all the scores are zero for points outside of U , we can only calculate them for $u \in U$.

We find the set Γ introduced in Lemma 4 and for every $\gamma \in \Gamma$ we construct $U_\gamma = U \cap (\mathcal{L} + \gamma)$. Consider any $u \in U_\gamma$ for some fixed $\gamma \in \Gamma$ and any $V \in \mathcal{V}$. We observe that if $V \not\equiv \gamma$, then $u \notin V$ and thus V does not contribute to $\text{score}(u)$. If $V \equiv \gamma$, then we can find a parquet W such that $V = W \cap (\mathcal{L} + \gamma)$ and we have $u \in V \Leftrightarrow u \in W \cap (\mathcal{L} + \gamma) \Leftrightarrow u \in W$. Thus, if we denote \mathcal{W}_γ as the set of parquets W obtained for every $V \in \mathcal{V}$ such that $V \equiv \gamma$, then $\text{score}(u)$ for $u \in U_\gamma$ is the number of parquets $W \in \mathcal{W}_\gamma$ such that $u \in W$. We calculate $\text{score}(u)$ for every $u \in U_\gamma$ by sweeping U_γ and \mathcal{W}_γ in time $\tilde{\mathcal{O}}(|U_\gamma| + |\mathcal{W}_\gamma|)$. We do it independently for every $\gamma \in \Gamma$, performing $\tilde{\mathcal{O}}(|U| + |\mathcal{V}|) = \tilde{\mathcal{O}}(n^2 + |\mathcal{V}|)$ operations in total.

Now consider a query vector $q \in Q$. Let $\gamma \in \Gamma$ be such that $q \equiv \gamma$. We have already showed that the sum of scores for $u \in D + q$ is equal to the sum of scores for $u \in (D + q) \cap U$. Since $(D + q) \cap U = (D + q) \cap U_\gamma$, we see that the result is the sum of scores for $u \in U_\gamma$ such that $\varphi \times u \geq \varphi \times q$ and $\psi \times u \geq \psi \times q$. If we denote $Q_\gamma = Q \cap (\mathcal{L} + \gamma)$, we see that we can calculate the results for all $q \in Q_\gamma$ by sweeping Q_γ and U_γ in time $\tilde{\mathcal{O}}(|Q_\gamma| + |U_\gamma|)$. We do it independently for every $\gamma \in \Gamma$, performing $\tilde{\mathcal{O}}(|Q| + |U|) = \tilde{\mathcal{O}}(n^2 + |Q|)$ operations in total. \square

Lemma 6. *For any simple subparquet U we can find $w_0, \dots, w_3 \in \mathbb{Z}^2$ such that*

$$|U \cap X| = \sum_{j=0}^3 (-1)^j |(D + w_j) \cap X|$$

for every $X \subseteq \mathbb{Z}^2$. If U consists of vectors of length $\mathcal{O}(n)$, then w_0, \dots, w_3 are of length $\mathcal{O}(n)$.

Proof. TODO \square

We apply Lemma 6 to every U_i and find $w_{i,0}, \dots, w_{i,3}$ so that we have

$$\begin{aligned} \sum_{i=1}^{\ell} |(U_i + q) \cap V_i| &= \sum_{i=1}^{\ell} |U_i \cap (V_i - q)| = \sum_{i=1}^{\ell} \sum_{j=0}^3 (-1)^j |(D + w_{i,j}) \cap (V_i - q)| \\ &= \sum_{j=0}^3 (-1)^j \sum_{i=1}^{\ell} |(D + q) \cap (V_i - w_{i,j})|. \end{aligned}$$

By Lemma 5 we can independently calculate the values $\sum_{i=1}^{\ell} |(D + q) \cap (V_i - w_{i,j})|$ for every j by running the algorithm for $\mathcal{V}_j = \{V_i - w_{i,j} : i \in \{1, \dots, \ell\}\}$ and Q .

4.5 Proof of Theorem 8

Definition 13 (Lattice graph). For a set $U \subseteq \mathbb{Z}^2$ we define its **lattice graph** $G_U = (U, E_U)$ where

$$E_U = \{ \{u, u + \delta\} : \delta \in \{\varphi, \psi\}, u \in U, u + \delta \in U \}$$

so every vector is connected with its translations by $\varphi, \psi, -\varphi, -\psi$.

Lemma 7. *If U is a spacious subparquet, then G_U is connected.*

Firstly, we partition R into a set of subparquet strings \mathcal{S} . For every non-empty $S \in \mathcal{S}$ we consider a lattice graph $G_{\text{dom}(S)}$. If S is not monochromatic, then since $G_{\text{dom}(S)}$ is connected, there must exist a pair of neighboring vectors v, w such that $S(v) \neq S(w)$. We select any such pair and partition S into spacious (or simple if S is simple) subparquet strings S' and S'' such that $v \in S'$ and $w \in S''$. For example if $v = w + \varphi$, then $S' = \{u : u \in S, \psi \times u \leq \psi \times v\}$ and $S'' = \{u : u \in S, \psi \times u > \psi \times v\}$. In the cases when $v = w + \delta$ for $\delta \in \{-\varphi, \psi, -\psi\}$ the construction is similar.

We can recursively partition S' and S'' further until we obtain monochromatic strings. Because R has $\mathcal{O}(k)$ -periods φ and ψ , the total number of neighbor pairs v, w such that $S(v) \neq S(w)$ is $\mathcal{O}(k)$ throughout all $S \in \mathcal{S}$. Thus the total number of recursive calls is $\mathcal{O}(k)$ and because $|\mathcal{S}| = \mathcal{O}(k)$, the total number of constructed strings is $\mathcal{O}(k)$. The algorithm can be implemented to work in time $\tilde{\mathcal{O}}(|\text{dom}(R)|)$.

4.6 Proof of Theorem 10

We assume ℓ to be an even number smaller than $\frac{n}{4}$ (if it is not, we can find $\ell' = \Theta(\ell)$, which is). We start by partitioning $\text{dom}(T)$ into **tiles**. We define $\varphi_{\min} = \min \{ \varphi \times u : u \in \text{dom}(T) \}$, analogously $\varphi_{\max}, \psi_{\min}, \psi_{\max}$ and denote $\delta_\varphi = \frac{\varphi_{\max} - \varphi_{\min}}{\ell}$, $\delta_\psi = \frac{\psi_{\max} - \psi_{\min}}{\ell}$. We define a tile with integer coordinates (s, t) as a set of vectors $u \in \mathbb{Z}^2$ such that

$$\varphi_{\min} + s\delta_\varphi < \varphi \times u \leq \varphi_{\min} + (s+1)\delta_\varphi,$$

$$\psi_{\min} + t\delta_\psi < \psi \times u \leq \psi_{\min} + (t+1)\delta_\psi.$$

For a fixed tile U consider $x_{\min} = \min \{ u.x : u \in U \}$, analogously $x_{\max}, y_{\min}, y_{\max}$ and a set

$$R = \{ u : u \in \mathbb{Z}^2, x_{\min} \leq u.x \leq x_{\max}, y_{\min} \leq u.y \leq y_{\max} \}.$$

We classify U into one of three types:

- a) if $U \cap T_{\mathbf{a}} = \emptyset$ then U is an inactive tile,
- b) if $U \cap T_{\mathbf{a}} \neq \emptyset$, $R \not\subseteq T_{\mathbf{a}}$ then U a border tile,
- c) if $U \cap T_{\mathbf{a}} \neq \emptyset$, $R \subseteq T_{\mathbf{a}}$ then U is an active tile.

We define B as a set of all $u \in T_{\mathbf{a}}$ contained in a border tile and construct $F = (B, T^{\mathbf{f}})$. Let us denote $z = \frac{n-1}{2}$. Consider a family of sets $\mathcal{R} = \{ R_i^1 \} \cup \{ R_i^2 \} \cup \{ R_i^3 \} \cup \{ R_i^4 \}$, where for every active tile U with coordinates (s, t) its members are placed into exactly one subset:

- 1) R_t^1 if $y_{\min} > z$, $x_{\max} \geq z$,
- 2) R_s^2 if $x_{\max} < z$, $y_{\max} \geq z$,
- 3) R_t^3 if $y_{\max} < z$, $x_{\min} \leq z$,
- 4) R_s^4 if $x_{\min} > z$, $y_{\min} \leq z$.

The number of non-empty sets $R \in \mathcal{R}$ is $\mathcal{O}(\ell)$. For each of them we consider $S = (R, T^{\mathbf{f}})$ which is a simple parquet string with $\mathcal{O}(k)$ -periods φ and ψ , and we further partition it using algorithm from Theorem 8, thus constructing the set \mathcal{U} .

4.7 Border proximity

In this section we explore the properties of strings defined for points close to the border. We consider any non-empty string S , such that $\text{dom}(S) \subseteq \text{dom}(T_{\mathbf{a}})$ and let $d = \text{BD}(S)$. We define a partitioning of S into strings S_1, \dots, S_4 , by splitting it through the middle with a horizontal and vertical line. Specifically

- S_1 is the restriction of S to $\{n/2, \dots, n-1\} \times \{n/2, \dots, n-1\}$ (upper right quarter),
- S_2 is the restriction of S to $\{0, \dots, n/2-1\} \times \{n/2, \dots, n-1\}$ (upper left quarter),
- S_3 is the restriction of S to $\{0, \dots, n/2-1\} \times \{0, \dots, n/2-1\}$ (lower left quarter),
- S_4 is the restriction of S to $\{n/2, \dots, n-1\} \times \{0, \dots, n/2-1\}$ (lower right quarter).

We will demonstrate some characteristics of S_1 , and by symmetry generalize them to S .

Lemma 8. *Assuming $d \leq m/4$, there does not exist $u \in S_1$ and $v \in T_{\mathbf{a}}$ such that $v.x - u.x \geq d$ and $v.y - u.y \geq d$.*

Proof. Assume the contrary. Since $u \in S_1$, we have $\text{BD}(u) \leq d$, so there exists $w \in \mathbb{Z}^2 \setminus \text{dom}(T_{\mathbf{a}})$ such that $u.x - d \leq w.x \leq u.x + d$ and $u.y - d \leq w.y \leq u.y + d$. Since $v \in T_{\mathbf{a}}$, there exists $q \in Q$ such that $v \in [m]^2 + q$. We have

$$w.x \geq u.x - d \geq n/2 - m/4 \geq n - m > q.x$$

and

$$w.x \leq u.x + d \leq v.x \leq q.x + m - 1.$$

Similarly we can show that $q.y \leq w.y \leq q.y + m - 1$, and thus $w \in [m]^2 + q$. Since $[m]^2 + q \subseteq \text{dom}(T_{\mathbf{a}})$ and $w \notin T_{\mathbf{a}}$, we get a contradiction. \square

Definition 14. (width & height) For a non-empty set $U \subseteq \mathbb{Z}^2$ we define its **width** as $\max\{u.x - v.x + 1 : u, v \in U\}$ and its **height** as $\max\{u.y - v.y + 1 : u, v \in U\}$. For a non-empty string R we define the width and height as the width and height of $\text{dom}(R)$.

Theorem 12. *Given two strings P, T of widths w_P, w_T and heights h_P, h_T , we can calculate $\text{Ham}(P + q, T)$ for every $q \in \mathbb{Z}^2$, for which the result is non-zero, in total time $\tilde{O}(|\Sigma|(w_P + w_T)(h_P + h_T))$, where $|\Sigma|$ denotes the number of different characters present in both P and T .*

Proof. We can prove it by slightly generalizing Theorem 4, although following the same method, and utilizing Observation 1. \square

Theorem 13. *We can calculate $\text{Ham}(P + q, S)$ for every $q \in Q$ in total time $\tilde{O}(m^2 + md|\Sigma|)$, where $|\Sigma|$ is the number of different characters present in both P and S .*

Proof. See Section 4.7.1. \square

4.7.1 Proof of Theorem 12

Recall that $\text{Ham}(P + q, S) = \text{Ham}(P + q, S_1) + \dots + \text{Ham}(P + q, S_4)$. We will only show how to calculate $\text{Ham}(P + q, S_1)$ for every $q \in Q$, since the other cases are symmetric.

We will take advantage of the fact that the points close to the border can overlap only with a small subset of points from the pattern when considering the occurrences fully contained in the active text. Specifically, let us consider a string P_0 defined as the restriction of P to $[n-d] \times [n-d]$ and a string P_1 defined as the restriction of P to $\text{dom}(P) \setminus \text{dom}(P_0)$. Since the strings P_0 and P_1 partition P , we have

$$\text{Ham}(P + q, S_1) = \text{Ham}(P_0 + q, S_1) + \text{Ham}(P_1 + q, S_1).$$

Lemma 9. $\text{Ham}(P_0 + q, S_1) = 0$ for every $q \in Q$.

Proof. We will show that $(P_0 + q) \cap S_1 = \emptyset$ for every $q \in Q$. Let us assume the contrary. We select any $u \in (P_0 + q) \cap S_1$ and consider a vector v such that $v.x = u.x + d$ and $v.y = u.y + d$. Since $u \in P_0 + q$, we have $v \in P + q \subseteq \text{dom}(T_a)$, thus u and v contradict Lemma 8. \square

Observation 3. P_1 can be partitioned into two strings P_2 and P_3 such that the width of P_2 and the height of P_3 are $\mathcal{O}(d)$.

By Lemma 9 and Observation 3 we have for every $q \in Q$

$$\text{Ham}(P + q, S_1) = \text{Ham}(P_1 + q, S_1) = \text{Ham}(P_2 + q, S_1) + \text{Ham}(P_3 + q, S_2).$$

We will only show how to calculate $\text{Ham}(P_2 + q, S_1)$, since the other case is symmetric.

4.7.2 Proof of Nothing

We will assume $d \leq m/4$, since for $d > m/4$ the proof is trivial. Consider an array of strings $U_0, \dots, U_{\lceil n/d \rceil - 1}$, where U_i is the restriction of S_1 to $\{id, \dots, id + d - 1\} \times [n] \cap \text{dom}(S_1)$. For the sake of formality (since the maximum/minimum of an empty set is undefined), let $V_0, \dots, V_{\ell-1}$ consist of all non-empty strings U_i , given in the ascending order of i . Observe that $V_0, \dots, V_{\ell-1}$ partition S_1 , their width is $\mathcal{O}(d)$ and it remains to show that the sum of their heights is $\mathcal{O}(m)$. Let us denote $a_i = \min \{u.y : u \in V_i\}$ and $b_i = \max \{u.y : u \in V_i\}$. Our goal is to prove that $\sum_{i=0}^{\ell-1} b_i - a_i + 1 = \mathcal{O}(m)$. Since for $\ell < 2$, the proof is trivial, we will assume $\ell \geq 2$.

Lemma 10. $b_{i+2} - a_i \leq d$ for every $i < \ell - 2$.

Proof. Let us assume the contrary and have $b_{i+2} - a_i > d$ for some i . There exists $u \in V_i$ such that $u.y = a_i$ and $v \in V_{i+2}$ such that $v.y = b_{i+2}$. We have $v.y - u.y > d$ (by assumption) and $v.x - u.x > d$ (since $u \in V_i, v \in V_{i+2}$), which contradicts Lemma 8. \square

By Lemma 10, we have

$$\sum_{i=0}^{\ell-1} b_i - a_i + 1 = \ell + b_0 + b_1 - a_{\ell-2} - a_{\ell-1} + \sum_{i=0}^{\ell-3} b_{i+2} - a_i \leq 3m + (\ell - 2) \cdot d = \mathcal{O}(m).$$

4.8 Proof of Theorem 11

Recall the construction of the sets \mathcal{V}_σ described in Section 4.1. We define $\sigma \in \Sigma$ to be a **frequent** character if $|\mathcal{V}_\sigma| \geq \sqrt{k}$ and if $|\mathcal{V}_\sigma| < \sqrt{k}$, we call it an **infrequent** character.

Observation 4. The number of different frequent characters is $\mathcal{O}(\sqrt{k})$.

We partition F into two strings S and R based on character frequency. We want S to consist of only the frequent characters and R to consist of only the infrequent ones. For every $q \in Q$ we then have

$$\text{Ham}(P + q, F) = \text{Ham}(P + q, S) + \text{Ham}(P + q, R).$$

Assuming S is non-empty, by Observation 4 and Theorem 12, we can calculate $\text{Ham}(P + q, S)$ for every $q \in Q$ in total time $\tilde{\mathcal{O}}(m^2 + mk^{1/2} \text{BD}(F))$, since $\text{BD}(S) \leq \text{BD}(F)$.

We now consider the calculation of $\text{Ham}(P + q, R)$ for every $q \in Q$.

4.8.1 Infrequent character contribution

References

- [1] Raphaël Clifford, Allyx Fontaine, Ely Porat, Benjamin Sach, and Tatiana Starikovskaya. The k-mismatch problem revisited. *CoRR*, abs/1508.00731, 2015.
- [2] Pawel Gawrychowski and Przemyslaw Uznanski. Optimal trade-offs for pattern matching with k mismatches. *CoRR*, abs/1704.01311, 2017.
- [3] Howard J. Karloff. Fast algorithms for approximately counting mismatches. *Inf. Process. Lett.*, 48(2):53–60, 1993.