Permutation Pattern Matching for Separable Permutations.

Both Emerite Neou[1], Romeo Rizzi [2], Stéphane Vialette[1]

Université Paris-Est, LIGM (UMR 8049), CNRS, UPEM, ESIEE Paris, ENPC, F-77454, Marne-la-Vallée, France
{neou,vialette}@univ-mlv.fr

Department of Computer Science, Università degli Studi di Verona, Italy
romeo.rizzi@univr.it

October 20, 2016

Plan

Permutation.

Permutation

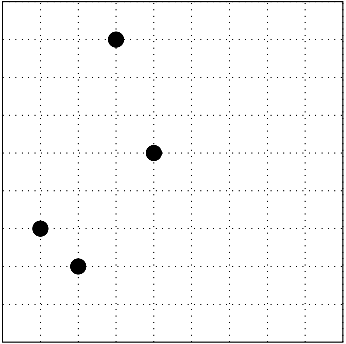
Two orders over a finite set.

- Usually the ordered set is a set of integers.
- Written as word $\pi = \pi[1]\pi[2]\ldots\pi[n]$.

Plot of a Permutation.

- We associate a figure to a permutation called a plot.
- Each element of a permutation is represented by the point $(i, \pi[i])$.

Example : the plot of the permutation 3 2 8 5.



Reduced form of a permutation and reduction.

Reduced Form of a Permutation

The elements are the first n integers.

Obtained by reducing a permutation.

- If π is on the set $\{e_1, e_2, \ldots, e_n\}$ where the naturall order is $e_1 < e_2 < \ldots < e_n$, we obtain the reduced form of π by renaming every e_i by i.

Example : reduced form of the permutation 3 2 8 5.

- $2 < 3 < 5 < 8$

Example : reduced form of the permutation 3 2 8 5.

- $2 < 3 < 5 < 8$

Example : reduced form of the permutation 3 2 8 5.

- $2 < 3 < 5 < 8$
- 2 becomes 1

Example : reduced form of the permutation 3 2 8 5.

- $2 < 3 < 5 < 8$
- 2 becomes 1
  3 becomes 2

Example : reduced form of the permutation 3 2 8 5.

- $2 < 3 < 5 < 8$
- 2 becomes 1
  3 becomes 2
  5 becomes 3

Example : reduced form of the permutation 3 2 8 5.

- $2 < 3 < 5 < 8$
- 2 becomes 1
  3 becomes 2
  5 becomes 3
  8 becomes 4

Example : reduced form of the permutation 3 2 8 5.

- $2 < 3 < 5 < 8$
- 2 becomes 1
  3 becomes 2
  5 becomes 3
  8 becomes 4
- 3 2 8 5 becomes 2 1 4 3

Occurrence

Occurrence

A mapping ϕ from a permutation pattern σ to a permutation text π is an occurrence of σ in π.

Occurrence

Occurrence

A mapping ϕ from a permutation pattern σ to a permutation text π is an occurrence of σ in π.

$$\Longleftrightarrow$$

The mapping is increasing and $\sigma[i] < \sigma[j]$ iff $\pi[\phi(i)] < \pi[\phi(j)]$.

Occurrence

Occurrence

A mapping ϕ from a permutation pattern σ to a permutation text π is an occurrence of σ in π.

$$\Longleftrightarrow$$

The mapping is increasing and $\sigma[i] < \sigma[j]$ iff $\pi[\phi(i)] < \pi[\phi(j)]$.

$$\Longleftrightarrow$$

The reduced form of the permutation given by the mapped elements is the same as the pattern.

Occurrence

Occurrence

A mapping ϕ from a permutation pattern σ to a permutation text π is an occurrence of σ in π.

$$\iff$$

The mapping is increasing and $\sigma[i] < \sigma[j]$ iff $\pi[\phi(i)] < \pi[\phi(j)]$.

$$\iff$$

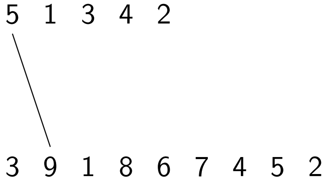
The reduced form of the permutation given by the mapped elements is the same as the pattern.

We say that σ occurs in π if such mapping exists.

Example : occurrence of 51342 in 391867452.

The mapping that map :

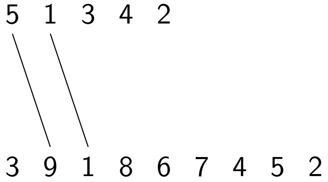
- 1 to 2,



Example : occurrence of 51342 in 391867452.

The mapping that map :

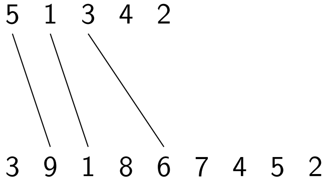
- 1 to 2,
- 2 to 3,



Example : occurrence of 51342 in 391867452.

The mapping that map :

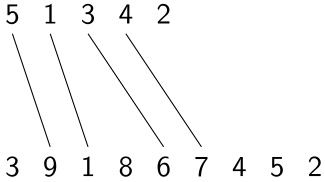
- 1 to 2,
- 2 to 3,
- 3 to 5,



Example : occurrence of 51342 in 391867452.

The mapping that map :

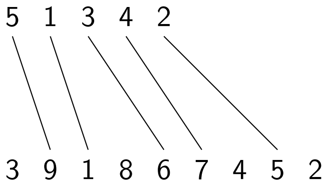
- 1 to 2,
- 2 to 3,
- 3 to 5,
- 4 to 6 and



Example : occurrence of 51342 in 391867452.

The mapping that map :

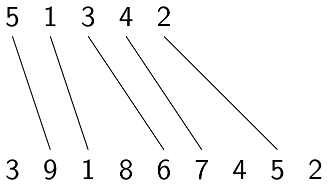
- 1 to 2,
- 2 to 3,
- 3 to 5,
- 4 to 6 and
- 5 to 8



Example : occurrence of 51342 in 391867452.

The mapping that map :

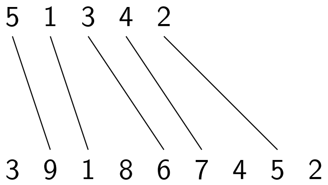
- 1 to 2,
- 2 to 3,
- 3 to 5,
- 4 to 6 and
- 5 to 8



Example : occurrence of 51342 in 391867452.

The mapping that map :

- 1 to 2,
- 2 to 3,
- 3 to 5,
- 4 to 6 and
- 5 to 8



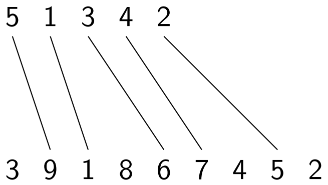
is an occurrence because:

- the i^{th} element of the mapping has the same position in the natural order than the i^{th} element in σ.

Example : occurrence of 51342 in 391867452.

The mapping that map :

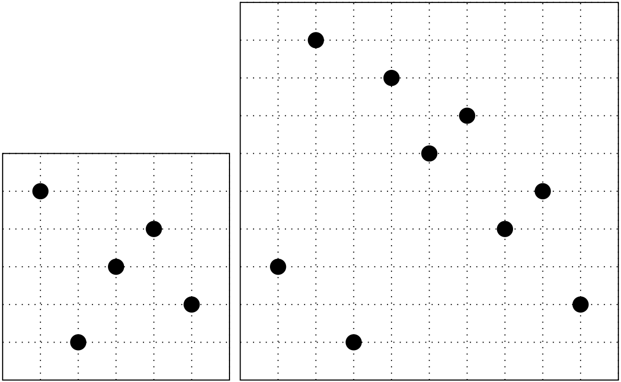
- 1 to 2,
- 2 to 3,
- 3 to 5,
- 4 to 6 and
- 5 to 8



is an occurrence because:

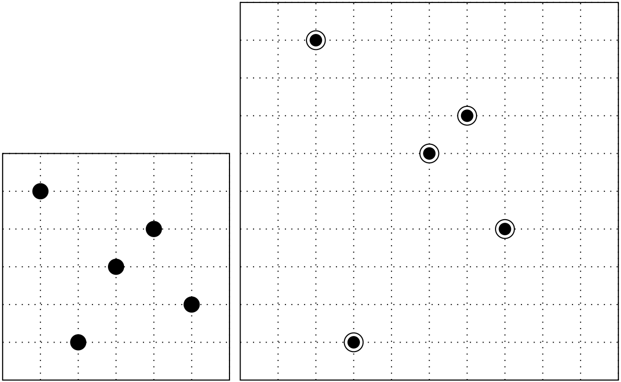
- the i^{th} element of the mapping has the same position in the natural order than the i^{th} element in σ.
- the permutation $\pi[2]\pi[3]\pi[5]\pi[6]\pi[8] = 91675$ is reduced to 51342.

Example : occurrence of 51342 in 391867452.



The plot of 51342 and 391867452.

Example : occurrence of 51342 in 391867452.



91674 is an occurrence.

Permutation Pattern Matching Problem.

Permutation Pattern Matching Problem

Given a pattern σ of size k and a text π of size n, we want to decide whether σ occurs in π.

Permutation Pattern Matching Problem.

The problem is NP-Complete (Bose et al. 1993). But

Permutation Pattern Matching Problem.

The problem is NP-Complete (Bose et al. 1993). But

- It can be solved in $O(1.79^{O(n)})$ (Ahal et al. 2008).

Permutation Pattern Matching Problem.

The problem is NP-Complete (Bose et al. 1993). But

- It can be solved in $O(1.79^{O(n)})$ (Ahal et al. 2008).
- It can be solved in $O(n.2^{O(k^2 \log k)})$: the problem is fixed-parameter tractable parameterized by the size of σ (Guillemot and Marx 2013).

Permutation Pattern Matching Problem.

The problem is NP-Complete (Bose et al. 1993). But

- It can be solved in $O(1.79^{O(n)})$ (Ahal et al. 2008).
- It can be solved in $O(n.2^{O(k^2 \log k)})$: the problem is fixed-parameter tractable parameterized by the size of σ (Guillemot and Marx 2013).
- Some variants of this problem are solved in polynomial time.

Variant for Permutation Pattern Matching Problem.

We can consider some variants for PPM-problem :

Variant for Permutation Pattern Matching Problem.

We can consider some variants for PPM-problem :

- Add constraint on the input text or/and pattern.
  $\Longrightarrow$ pattern or/and text is/are in a class.

Variant for Permutation Pattern Matching Problem.

We can consider some variants for PPM-problem :

- Add constraint on the input text or/and pattern.
  $\Longrightarrow$ pattern or/and text is/are in a class.
- Add constraint on the occurrence.
  $\Longrightarrow$ bivincular permutation pattern and mesh permutation pattern.

Our interests.

What we study :

Our interests.

What we study :

- What can we do if the pattern is a separable permutation?

Our interests.

What we study :

- What can we do if the pattern is a separable permutation?
- What can we do if the pattern and the text are separable permutation?

Our interests.

What we study :

- What can we do if the pattern is a separable permutation?
- What can we do if the pattern and the text are separable permutation?
- What can we do if the pattern is a bivincular separable permutation pattern?

Results.

- the pattern is separable:

Results.

- the pattern is separable:
  - best result : $O(kn^4)$ time and $O(kn^3)$ space algorithm.

Results.

- the pattern is separable:
  - best result : $O(kn^4)$ time and $O(kn^3)$ space algorithm.
  - our contribution : $O(kn^4)$ time and $O(\log(k)n^3)$ space algorithm.

Results.

- the pattern is separable:
  - best result : $O(kn^4)$ time and $O(kn^3)$ space algorithm.
  - our contribution : $O(kn^4)$ time and $O(\log(k)n^3)$ space algorithm.
- the pattern and the text are separable:

Results.

- the pattern is separable:
  - best result : $O(kn^4)$ time and $O(kn^3)$ space algorithm.
  - our contribution : $O(kn^4)$ time and $O(\log(k)n^3)$ space algorithm.
- the pattern and the text are separable:
  - best result :

$$O\left(\min\left\{\begin{array}{l} l_{T'}\, n_T \\ l_{T'}\, l_T \log\log n_T + n_T \\ \frac{n_T\, n_{T'}}{\log n_T} + n_T \log n_T \end{array}\right\}\right)$$

time and $O(n_T)$ space algorithm.

Results.

- the pattern is separable:
  - best result : $O(kn^4)$ time and $O(kn^3)$ space algorithm.
  - our contribution : $O(kn^4)$ time and $O(\log(k)n^3)$ space algorithm.
- the pattern and the text are separable:
  - best result :

$$O\left(\min\left\{\begin{array}{l} l_{T'}\, n_T \\ l_{T'}\, l_T \log\log n_T + n_T \\ \frac{n_T\, n_{T'}}{\log n_T} + n_T \log n_T \end{array}\right\}\right)$$

    time and $O(n_T)$ space algorithm.
  - our contribution :$O(n^2k)$ time and $O(nk)$ space algorithm.

Results.

- the pattern is separable:
  - best result : $O(kn^4)$ time and $O(kn^3)$ space algorithm.
  - our contribution : $O(kn^4)$ time and $O(\log(k)n^3)$ space algorithm.
- the pattern and the text are separable:
  - best result :

$$O\left(\min\left\{\begin{array}{l} l_{T'}\, n_T \\ l_{T'}\, l_T \log\log n_T + n_T \\ \frac{n_T\, n_{T'}}{\log n_T} + n_T \log n_T \end{array}\right\}\right)$$

    time and $O(n_T)$ space algorithm.
  - our contribution :$O(n^2k)$ time and $O(nk)$ space algorithm.
- the pattern is a bivincular separable pattern:

Results.

- the pattern is separable:
  - best result : $O(kn^4)$ time and $O(kn^3)$ space algorithm.
  - our contribution : $O(kn^4)$ time and $O(\log(k)n^3)$ space algorithm.
- the pattern and the text are separable:
  - best result :

$$O\left(\min\left\{\begin{array}{l} l_{T'}\, n_T \\ l_{T'}\, l_T \log\log n_T + n_T \\ \frac{n_T\, n_{T'}}{\log n_T} + n_T \log n_T \end{array}\right\}\right)$$

    time and $O(n_T)$ space algorithm.
  - our contribution :$O(n^2k)$ time and $O(nk)$ space algorithm.
- the pattern is a bivincular separable pattern:
  - $\emptyset$

Results.

- the pattern is separable:
  - best result : $O(kn^4)$ time and $O(kn^3)$ space algorithm.
  - our contribution : $O(kn^4)$ time and $O(\log(k)n^3)$ space algorithm.
- the pattern and the text are separable:
  - best result :

$$O\left(\min\left\{\begin{array}{l} l_{T'}\, n_T \\ l_{T'}\, l_T \log\log n_T + n_T \\ \frac{n_T\, n_{T'}}{\log n_T} + n_T \log n_T \end{array}\right\}\right)$$

    time and $O(n_T)$ space algorithm.
  - our contribution :$O(n^2 k)$ time and $O(nk)$ space algorithm.
- the pattern is a bivincular separable pattern:
  - $\emptyset$
  - $O(n^6 k)$ time and space algorithm.

Results.

- Longest common separable permutation with at least one separable permutation:

Results.

- Longest common separable permutation with at least one separable permutation:
  - best result : $O(n^8)$ time and space algorithm.

Results.

- Longest common separable permutation with at least one separable permutation:
  - best result : $O(n^8)$ time and space algorithm.
  - our contribution : $O(n^6 k)$ time and $O(n^4 \log k)$ space algorithm.

Results.

- Longest common separable permutation with at least one separable permutation:
  - best result : $O(n^8)$ time and space algorithm.
  - our contribution : $O(n^6 k)$ time and $O(n^4 \log k)$ space algorithm.
- Unshuffling of a permutation into two separable patterns:

Results.

- Longest common separable permutation with at least one separable permutation:
  - best result : $O(n^8)$ time and space algorithm.
  - our contribution : $O(n^6 k)$ time and $O(n^4 \log k)$ space algorithm.
- Unshuffling of a permutation into two separable patterns:
  - $\emptyset$

Results.

- Longest common separable permutation with at least one separable permutation:
  - best result : $O(n^8)$ time and space algorithm.
  - our contribution : $O(n^6 k)$ time and $O(n^4 \log k)$ space algorithm.
- Unshuffling of a permutation into two separable patterns:
  - $\emptyset$
  - our contribution : $O(nk^3\ell^2)$ algorithm.

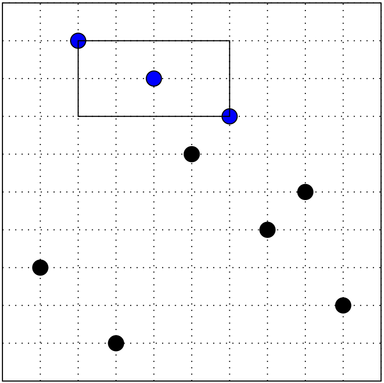
Plan

Rectangle

Recangle of a Permutation

A rectangle is a subsequence of the permutation, define as the subsequence of elements that are between a certain range of index, and the elements are not bigger than a given value and the elements are not smaller than a given value.

A rectangle in 391867452.



The rectangle is the permutation 987.

Direct Sum and Skew Sum

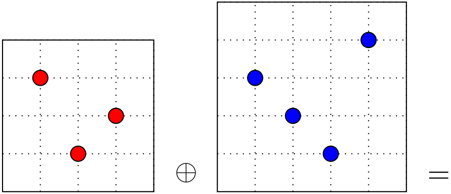
Direct Sum

Given π_1 of size n_1, π_2 of size n_2, $\pi_1 \oplus \pi_2 =$
$\pi_1[1]\pi_1[2]\ldots\pi_1[n_1](\pi_2[1]+n_1)(\pi_2[2]+n_1)\ldots(\pi_2[n_2]+n_1)$.

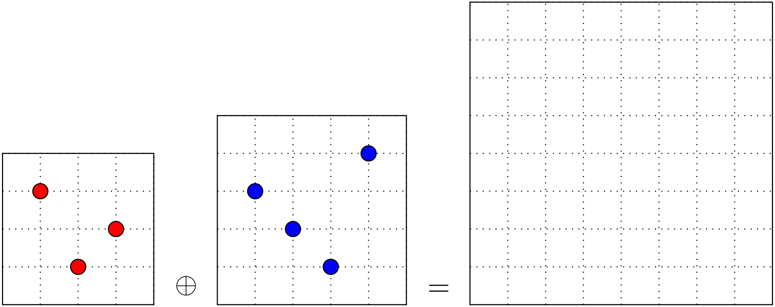
Skew Sum

Given π_1 of size n_1, π_2 of size n_2, $\pi_1 \ominus \pi_2 =$
$(\pi_1[1]+n_2)(\pi_1[2]+n_2)\ldots(\pi_1[n_1]+n_2)\pi_2[1]\pi_2[2]\ldots\pi_2[n_2]$.

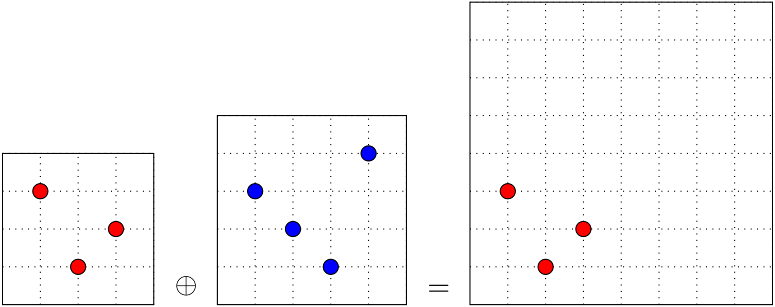
Example : Direct Sum



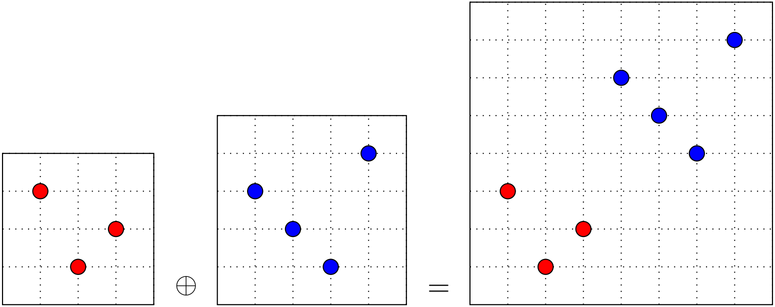
Example : Direct Sum



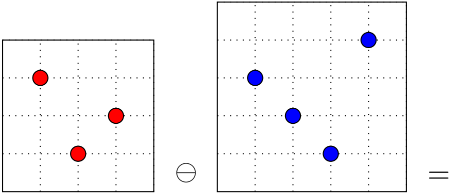
Example : Direct Sum



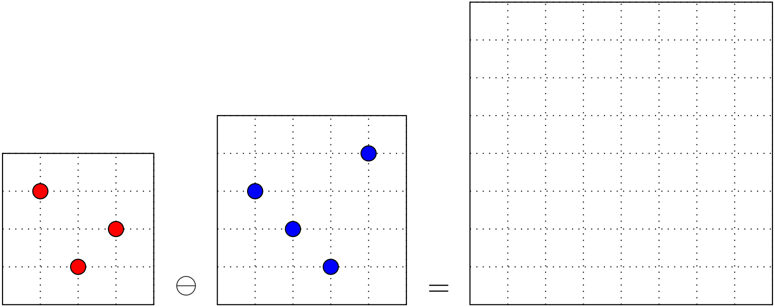
Example : Direct Sum



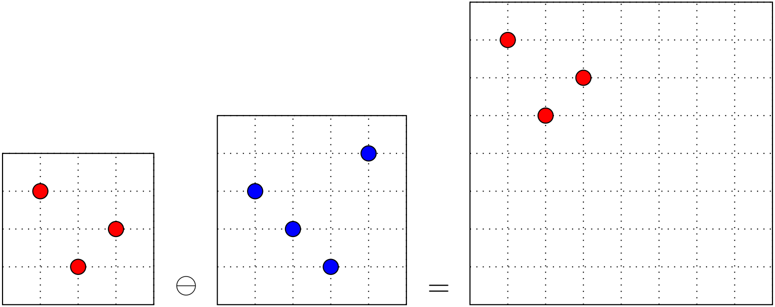
Example : Skew Sum



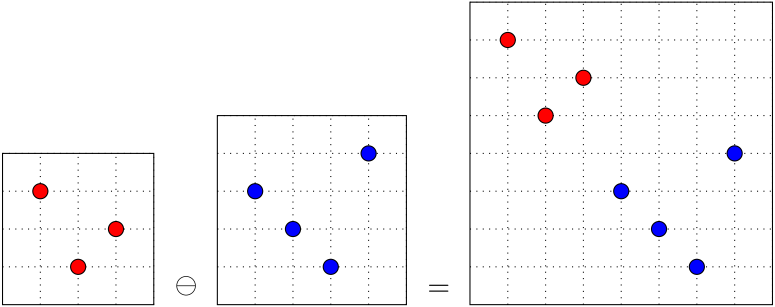
Example : Skew Sum



Example : Skew Sum



Example : Skew Sum



Separable Permutationn.

Separable Permutation

π is a separable permutation

Separable Permutationn.

Separable Permutation

π is a separable permutation

$$\Longleftrightarrow$$

$\pi = 1$
OR
$\pi = \pi_1 \oplus \pi_2$ or $\pi = \pi_1 \ominus \pi_2$, with π_1 and π_2 are separable permutation

Separable Permutationn.

Separable Permutation

π is a separable permutation

$$\Longleftrightarrow$$

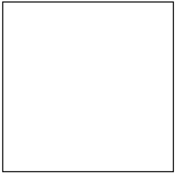
$\pi = 1$

OR

$\pi = \pi_1 \oplus \pi_2$ or $\pi = \pi_1 \ominus \pi_2$, with π_1 and π_2 are separable permutation

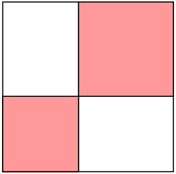
Example : decomposition of 214376589

- 214376589.



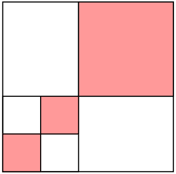
Example : decomposition of 214376589

- 214376589.
- $red(2143) \oplus red(76589) = 2143 \oplus 32145$.



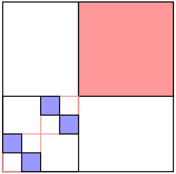
Example : decomposition of 214376589

- 214376589.
- $red(2143) \oplus red(76589) = 2143 \oplus 32145.$
- $2143 = red(21) \oplus red(43) = 21 \oplus 21.$



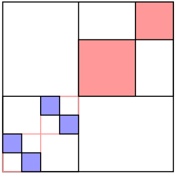
Example : decomposition of 214376589

- 214376589.
- $red(2143) \oplus red(76589) = 2143 \oplus 32145$.
- $2143 = red(21) \oplus red(43) = 21 \oplus 21$.
- $21 = red(2) \ominus red(1) = 1 \ominus 1$.



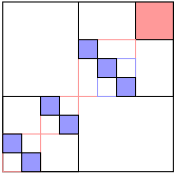
Example : decomposition of 214376589

- 214376589.
- $red(2143) \oplus red(76589) = 2143 \oplus 32145.$
- $2143 = red(21) \oplus red(43) = 21 \oplus 21.$
- $21 = red(2) \ominus red(1) = 1 \ominus 1.$
- $32145 = red(321) \oplus red(45) = 321 \oplus 12.$



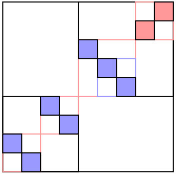
Example : decomposition of 214376589

- 214376589.
- $red(2143) \oplus red(76589) = 2143 \oplus 32145$.
- $2143 = red(21) \oplus red(43) = 21 \oplus 21$.
- $21 = red(2) \ominus red(1) = 1 \ominus 1$.
- $32145 = red(321) \oplus red(45) = 321 \oplus 12$.
- $765 = red(7) \ominus (red(6) \ominus red(5)) = (1 \ominus (1 \ominus 1))$.



Example : decomposition of 214376589

- 214376589.
- $red(2143) \oplus red(76589) = 2143 \oplus 32145.$
- $2143 = red(21) \oplus red(43) = 21 \oplus 21.$
- $21 = red(2) \ominus red(1) = 1 \ominus 1.$
- $32145 = red(321) \oplus red(45) = 321 \oplus 12.$
- $765 = red(7) \ominus (red(6) \ominus red(5)) = (1 \ominus (1 \ominus 1)).$
- $89 = red(8) \oplus red(9) = 1 \oplus 1.$



Plan

Dynamic Programming

Dynamic Programming :

- To solve an instance of the problem, we only need to solve smaller and easier instances of the problem, until the instances become trivial.

Trivial Case

The pattern 1 occurs in a rectangle if and only if the rectangle is not empty.

Algorithm

To decide whether $\sigma = \sigma_1 \oplus \sigma_2$ occurs in a rectangle R:

Algorithm

To decide whether $\sigma = \sigma_1 \oplus \sigma_2$ occurs in a rectangle R:

For every pair of rectangles R_1 and R_2 such that R_1 is left below R_2 :

Algorithm

To decide whether $\sigma = \sigma_1 \oplus \sigma_2$ occurs in a rectangle R:

For every pair of rectangles R_1 and R_2 such that R_1 is left below R_2 :
decide whether σ_1 occurs in R_1

Algorithm

To decide whether $\sigma = \sigma_1 \oplus \sigma_2$ occurs in a rectangle R:

For every pair of rectangles R_1 and R_2 such that R_1 is left below R_2 :
    decide whether σ_1 occurs in R_1
    decide whether σ_2 occurs in R_2

Algorithm

To decide whether $\sigma = \sigma_1 \oplus \sigma_2$ occurs in a rectangle R:

For every pair of rectangles R_1 and R_2 such that R_1 is left below R_2 :
    decide whether σ_1 occurs in R_1
    decide whether σ_2 occurs in R_2
    conclude that σ occurs in R.

Improvement of the Core

- We do not need to check every pair of rectangles

Improvement of the Core

- We do not need to check every pair of rectangles
- We do not need every edge of a rectangle.

Conclusion

Open problems:

- Separable permutation and mesh pattern?

Conclusion

Open problems:

- Separable permutation and mesh pattern?
- Av(321) and PPM?

Thank You!