Linear-Time Computation of Shortest Covers of All Rotations of a String

Maxime Crochemore[1,2], Costas S. Iliopoulos[1], Jakub Radoszewski[3]
Wojciech Rytter[3], Juliusz Straszyński[3], Tomasz Waleń[3], Wiktor Zuba[3,4]

1 King's College London, UK
2 Université Gustave Eiffel, Marne-la-Vallée, France
3 University of Warsaw, Poland
4 CWI, Amsterdam, The Netherlands

Cover

Definition (Cover, Quasiperiod)

A string C is a cover of a string S if the occurrences of C in S cover S.

Cover

Definition (Cover, Quasiperiod)

A string C is a cover of a string S if the occurrences of C in S cover S.

`a a b a a b a a a b a a a a b a a`

Quasiperiodicity:

- Introduced by Apostolico and Ehrenfeucht [TCS 93]
- Captures repetitive structures in strings defying classical periods

Cover

Definition (Cover, Quasiperiod)

A string C is a cover of a string S if the occurrences of C in S cover S.

`a a b a a b a a a b a a a a b a a`

Quasiperiodicity:

- Introduced by Apostolico and Ehrenfeucht [TCS 93]
- Captures repetitive structures in strings defying classical periods

Let S be a string of length n.

All the following can be computed in $\mathcal{O}(n)$ time:

- Shortest cover of S [Apostolico, Farach and Iliopoulos, IPL 91]
- Shortest covers of all prefixes of S [Breslauer, IPL 92]
- All covers of S [Moore and Smyth, IPL 95]

Shortest Covers of All Rotations

Definition (Rotation)

If $S = UV$ and $T = VU$, then T is a rotation of S ($T = rot_{|U|}(S)$).

Theorem (Crochemore+ [CIRRSWZ, TCS 21])

Shortest covers of all rotations of S can be computed in $\mathcal{O}(n \log n)$ time and $\mathcal{O}(n)$ space.

i	$rot_i(S)$	shortest cover	$CC[i]$
0	abaababa	aba	3
1	baababaa	baababaa	8
2	baababaa	aababaab	8
3	ababaaba	aba	3
4	babaabaa	babaabaa	8
5	abaabaab	abaab	5
6	baabaaba	baaba	5
7	aabaabab	aabaabab	8

Shortest Covers of All Rotations

Definition (Rotation)

If $S = UV$ and $T = VU$, then T is a rotation of S ($T = rot_{|U|}(S)$).

Theorem (Crochemore+ [CIRRSWZ, TCS 21])

Shortest covers of all rotations of S can be computed in $\mathcal{O}(n \log n)$ time and $\mathcal{O}(n)$ space.

	i	$rot_i(S)$	shortest cover	$CC[i]$	
Input	0	abaababa	aba	3	
	1	baababaa	baababaa	8	
	2	baababaa	aababaab	8	
	3	ababaaba	aba	3	
	4	babaabaa	babaabaa	8	
	5	abaabaab	abaab	5	
	6	baabaaba	baaba	5	Output
	7	aabaabab	aabaabab	8	

Shortest Covers of All Rotations

Definition (Rotation)

If $S = UV$ and $T = VU$, then T is a rotation of S ($T = rot_{|U|}(S)$).

Theorem (Crochemore+ [CIRRSWZ, TCS 21])

Shortest covers of all rotations of S can be computed in $\mathcal{O}(n \log n)$ time and $\mathcal{O}(n)$ space.

	i	$rot_i(S)$	shortest cover	$CC[i]$	
Input	0	abaababa	aba	3	
	1	baababaa	baababaa	8	
	2	baababaa	aababaab	8	
	3	ababaaba	aba	3	
	4	babaabaa	babaabaa	8	
	5	abaabaab	abaab	5	
	6	baabaaba	baaba	5	Output
	7	aabaabab	aabaabab	8	

- Is $\mathcal{O}(n)$ time possible?

Shortest Covers of All Rotations

Definition (Rotation)

If $S = UV$ and $T = VU$, then T is a rotation of S ($T = rot_{|U|}(S)$).

Theorem (Crochemore+ [CIRRSWZ, TCS 21])

Shortest covers of all rotations of S can be computed in $\mathcal{O}(n \log n)$ time and $\mathcal{O}(n)$ space.

	i	$rot_i(S)$	shortest cover	$CC[i]$	
Input	0	abaababa	aba	3	
	1	baababaa	baababaa	8	
	2	baababaa	aababaab	8	
	3	ababaaba	aba	3	
	4	babaabaa	babaabaa	8	
	5	abaabaab	abaab	5	
	6	baabaaba	baaba	5	Output
	7	aabaabab	aabaabab	8	

- Is $\mathcal{O}(n)$ time possible? [CIRRSWZ]
- $\min_i CC[i]$ can be computed in $\mathcal{O}(n)$ time for S over integer alphabet $[0, n^{\mathcal{O}(1)}]$

Motivation 1: Internal Shortest Covers

Definition (Internal shortest cover queries)

Compute the length of the shortest cover of a text factor.

`a b a a b a b a a b a b a a b a`

Motivation 1: Internal Shortest Covers

Definition (Internal shortest cover queries)

Compute the length of the shortest cover of a text factor.

a b a a b a b a a b a b a a b a

Motivation 1: Internal Shortest Covers

Definition (Internal shortest cover queries)

Compute the length of the shortest cover of a text factor.

a b a a b a b a a b a b a a b a

Theorem ([CIRRSWZ, SPIRE 20], [Belazzougui+, CPM 21])

Internal shortest cover queries in S can be answered in $\mathcal{O}(\log n)$ time after $\mathcal{O}(n \log n)$ preprocessing.

Motivation 1: Internal Shortest Covers

Definition (Internal shortest cover queries)

Compute the length of the shortest cover of a text factor.

a b a a b a b a a b a b a a b a

Theorem ([CIRRSWZ, SPIRE 20], [Belazzougui+, CPM 21])

Internal shortest cover queries in S can be answered in $\mathcal{O}(\log n)$ time after $\mathcal{O}(n \log n)$ preprocessing.

Alternative $\mathcal{O}(n \log n)$-time algorithm for shortest covers of all rotations:

b a a b a a b a

Motivation 1: Internal Shortest Covers

Definition (Internal shortest cover queries)

Compute the length of the shortest cover of a text factor.

a b a a b a b a a b a b a a b a

Theorem ([CIRRSWZ, SPIRE 20], [Belazzougui+, CPM 21])

Internal shortest cover queries in S can be answered in $\mathcal{O}(\log n)$ time after $\mathcal{O}(n \log n)$ preprocessing.

Alternative $\mathcal{O}(n \log n)$-time algorithm for shortest covers of all rotations:

b a a b a a b a | b a a b a a b a

Motivation 1: Internal Shortest Covers

Definition (Internal shortest cover queries)

Compute the length of the shortest cover of a text factor.

a b a a b a b a a b a b a a b a

Theorem ([CIRRSWZ, SPIRE 20], [Belazzougui+, CPM 21])

Internal shortest cover queries in S can be answered in $\mathcal{O}(\log n)$ time after $\mathcal{O}(n \log n)$ preprocessing.

Alternative $\mathcal{O}(n \log n)$-time algorithm for shortest covers of all rotations:

b a a b a a b a | b a a b a a b a

Motivation 1: Internal Shortest Covers

Definition (Internal shortest cover queries)

Compute the length of the shortest cover of a text factor.

a b a a b a b a a b a b a a b a

Theorem ([CIRRSWZ, SPIRE 20], [Belazzougui+, CPM 21])

Internal shortest cover queries in S can be answered in $\mathcal{O}(\log n)$ time after $\mathcal{O}(n \log n)$ preprocessing.

Alternative $\mathcal{O}(n \log n)$-time algorithm for shortest covers of all rotations:

b a a b a a b a|b a a b a a b a

Motivation 1: Internal Shortest Covers

Definition (Internal shortest cover queries)

Compute the length of the shortest cover of a text factor.

a b a a b a b a a b a b a a b a

Theorem ([CIRRSWZ, SPIRE 20], [Belazzougui+, CPM 21])

Internal shortest cover queries in S can be answered in $\mathcal{O}(\log n)$ time after $\mathcal{O}(n \log n)$ preprocessing.

Alternative $\mathcal{O}(n \log n)$-time algorithm for shortest covers of all rotations:

b a a b a a b a | b a a b a a b a

Motivation 1: Internal Shortest Covers

Definition (Internal shortest cover queries)

Compute the length of the shortest cover of a text factor.

a **b a a b a b a a b a b a** a b a

Theorem ([CIRRSWZ, SPIRE 20], [Belazzougui+, CPM 21])

Internal shortest cover queries in S can be answered in $\mathcal{O}(\log n)$ time after $\mathcal{O}(n \log n)$ preprocessing.

Alternative $\mathcal{O}(n \log n)$-time algorithm for shortest covers of all rotations:

b a a **b a a b a | b a a** b a a b a

Motivation 1: Internal Shortest Covers

Definition (Internal shortest cover queries)

Compute the length of the shortest cover of a text factor.

a b a a b a b a a b a b a a b a

Theorem ([CIRRSWZ, SPIRE 20], [Belazzougui+, CPM 21])

Internal shortest cover queries in S can be answered in $\mathcal{O}(\log n)$ time after $\mathcal{O}(n \log n)$ preprocessing.

Alternative $\mathcal{O}(n \log n)$-time algorithm for shortest covers of all rotations:

b a a b a a b a | b a a b a a b a

Motivation 1: Internal Shortest Covers

Definition (Internal shortest cover queries)

Compute the length of the shortest cover of a text factor.

a b a a b a b a a b a b a a b a

Theorem ([CIRRSWZ, SPIRE 20], [Belazzougui+, CPM 21])

Internal shortest cover queries in S can be answered in $\mathcal{O}(\log n)$ time after $\mathcal{O}(n \log n)$ preprocessing.

Alternative $\mathcal{O}(n \log n)$-time algorithm for shortest covers of all rotations:

b a a b a a b a | b a a b a a b a

Motivation 2: Seeds

Definition (Seed)

A string C is a seed of a string S if C is a cover of a superstring of S.

`a b a a b a a a b a a a a b`

Motivation 2: Seeds

Definition (Seed)

A string C is a seed of a string S if C is a cover of a superstring of S.

a a b a a b a a a b a a a a b a a

Motivation 2: Seeds

Definition (Seed)

A string C is a seed of a string S if C is a cover of a superstring of S.

`a b a a b a a a b a a a a b`

In other words, S is covered by occurrences and left/right overhangs of C.

Motivation 2: Seeds

Definition (Seed)

A string C is a seed of a string S if C is a cover of a superstring of S.

`a b a a b a a a b a a a a b`

In other words, S is covered by occurrences and left/right overhangs of C.

An $\mathcal{O}(n)$-sized representation of all seeds of S can be computed:

- in $\mathcal{O}(n \log n)$ time [Iliopoulos, Moore, Park, Algorithmica 96]
- in $\mathcal{O}(n)$ time over integer alphabet, Kociumaka+ [KKRRW, SODA 12]

Seeds in [CIRRSWZ, TCS 21]

S

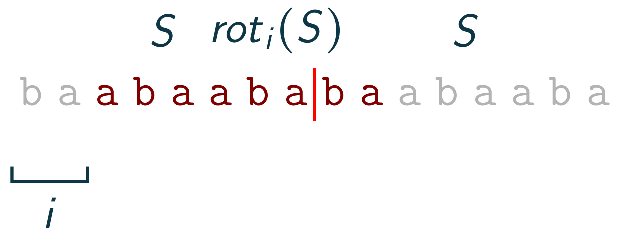
b a a b a a b a

Seeds in [CIRRSWZ, TCS 21]

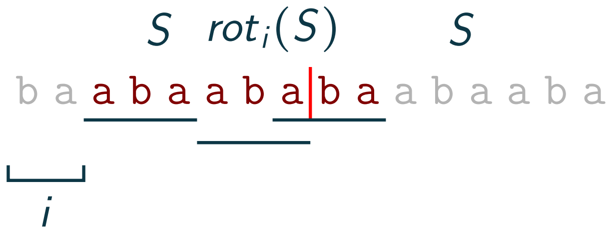
S S

b a a b a a b a | b a a b a a b a

Seeds in [CIRRSWZ, TCS 21]

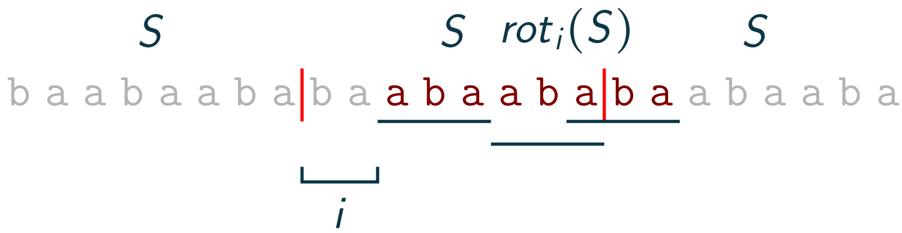


Seeds in [CIRRSWZ, TCS 21]



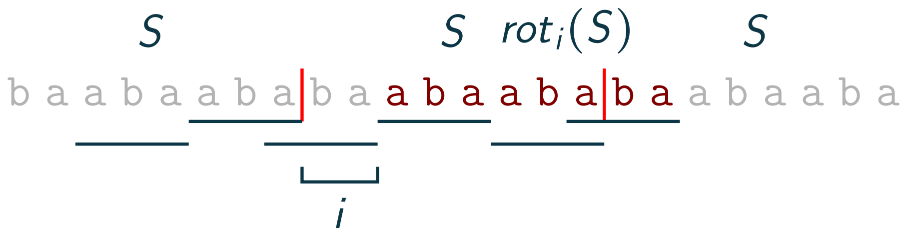
C is a cover of $rot_i(S)$

Seeds in [CIRRSWZ, TCS 21]



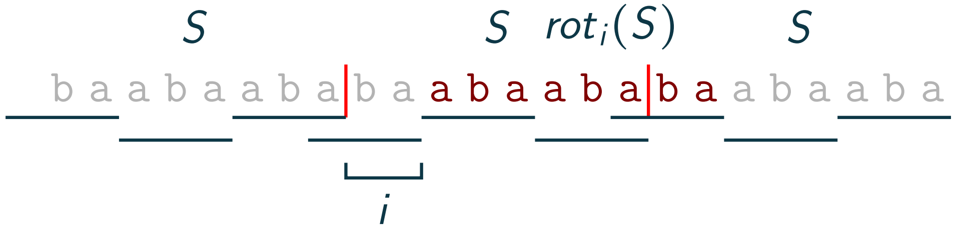
C is a cover of $rot_i(S)$

Seeds in [CIRRSWZ, TCS 21]



C is a cover of $rot_i(S)$

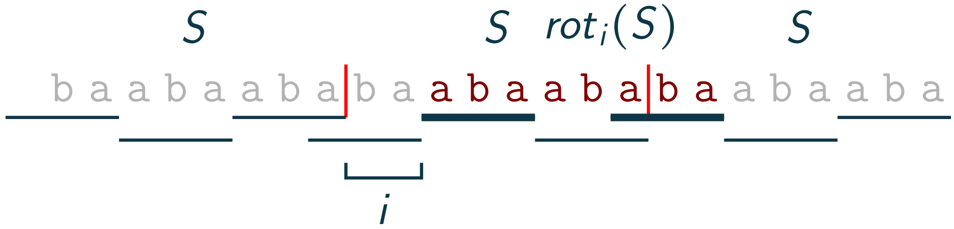
Seeds in [CIRRSWZ, TCS 21]



If C is a cover of $rot_i(S)$,

then C is a seed of S^3

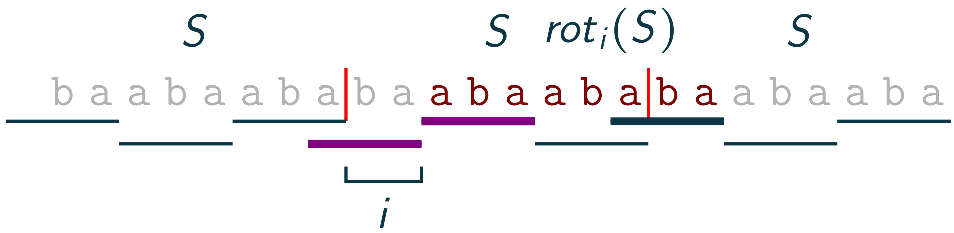
Seeds in [CIRRSWZ, TCS 21]



If C is a cover of $rot_i(S)$,

then C is a seed of S^3

Seeds in [CIRRSWZ, TCS 21]

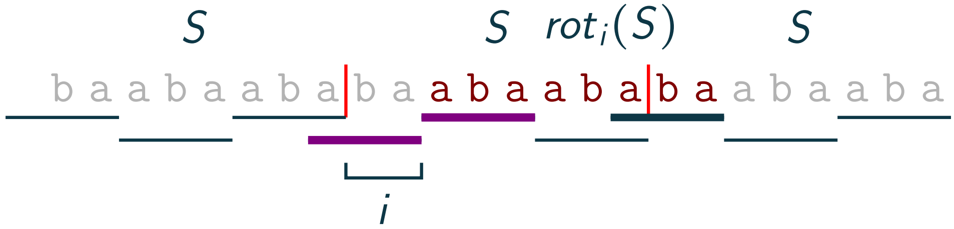


If C is a cover of $rot_i(S)$,

then C is a seed of S^3

and C^2 occurs in S^3 with center at position $n + i$.

Seeds in [CIRRSWZ, TCS 21]



If C is a cover of $rot_i(S)$,

then C is a seed of S^3

and C^2 occurs in S^3 with center at position $n + i$.

Converse is also true if $|C| \leq |S|$.

Seeds in [CIRRSWZ, TCS 21]

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then C is a cover of $rot_i(S)$ iff C is a seed of T and C^2 occurs in T with center at position $n + i$.

Seeds in [CIRRSWZ, TCS 21]

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then C is a cover of $rot_i(S)$ iff C is a seed of T and C^2 occurs in T with center at position $n+i$.

String C is primitive if C is not a (full) power of a shorter string.

Observation. The shortest cover of a string is primitive.

Seeds in [CIRRSWZ, TCS 21]

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then C is a cover of $rot_i(S)$ iff C is a seed of T and C^2 occurs in T with center at position $n + i$.

String C is primitive if C is not a (full) power of a shorter string.

Observation. The shortest cover of a string is primitive.

Proof. If $C = U^k$ is a cover of S, then U is also a cover of S.

Seeds in [CIRRSWZ, TCS 21]

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then C is a cover of $rot_i(S)$ iff C is a seed of T and C^2 occurs in T with center at position $n + i$.

String C is primitive if C is not a (full) power of a shorter string.

Observation. The shortest cover of a string is primitive.

Seeds in [CIRRSWZ, TCS 21]

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then C is a cover of $rot_i(S)$ iff C is a seed of T and C^2 occurs in T with center at position $n+i$.

String C is primitive if C is not a (full) power of a shorter string.

Observation. The shortest cover of a string is primitive.

U^2 is a p-square (primitively-rooted square) if U is primitive.

Seeds in [CIRRSWZ, TCS 21]

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then C is a cover of $rot_i(S)$ iff C is a seed of T and C^2 occurs in T with center at position $n + i$.

String C is primitive if C is not a (full) power of a shorter string.

Observation. The shortest cover of a string is primitive.

U^2 is a p-square (primitively-rooted square) if U is primitive.

- **PrimCov**$[i]$: primitive covers of $rot_i(S)$
- **Seeds**(T): seeds of T
- **PSquares**(T): p-square factors of T
- **Centers**(U^2, T): centers of occurrences of U^2 in T

Seeds in [CIRRSWZ, TCS 21]

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then C is a cover of $rot_i(S)$ iff C is a seed of T and C^2 occurs in T with center at position $n + i$.

String C is primitive if C is not a (full) power of a shorter string.

Observation. The shortest cover of a string is primitive.

U^2 is a p-square (primitively-rooted square) if U is primitive.

- **PrimCov**$[i]$: primitive covers of $rot_i(S)$
- **Seeds**(T): seeds of T
- **PSquares**(T): p-square factors of T
- **Centers**(U^2, T): centers of occurrences of U^2 in T

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then $C \in \textbf{PrimCov}[i]$ iff $C \in \textbf{Seeds}(T)$, $C^2 \in \textbf{PSquares}(T)$ and $n + i \in \textbf{Centers}(C^2, T)$.

Seeds in [CIRRSWZ, TCS 21]

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then C is a cover of $rot_i(S)$ iff C is a seed of T and C^2 occurs in T with center at position $n + i$.

String C is primitive if C is not a (full) power of a shorter string.

Observation. The shortest cover of a string is primitive.

U^2 is a p-square (primitively-rooted square) if U is primitive.

- **PrimCov**$[i]$: primitive covers of $rot_i(S)$
- **Seeds**(T): seeds of T
- **PSquares**(T): p-square factors of T
- **Centers**(U^2, T): centers of occurrences of U^2 in T

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in$ **PrimCov**$[i]$ iff

$C \in$ **Seeds**(T), $C^2 \in$ **PSquares**(T) and $n + i \in$ **Centers**(C^2, T).

Seeds in [CIRRSWZ, TCS 21]

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then C is a cover of $rot_i(S)$ iff C is a seed of T and C^2 occurs in T with center at position $n + i$.

String C is primitive if C is not a (full) power of a shorter string.

Observation. The shortest cover of a string is primitive.

U^2 is a p-square (primitively-rooted square) if U is primitive.

- **PrimCov**$[i]$: primitive covers of $rot_i(S)$
- **Seeds**(T): seeds of T
- **PSquares**(T): p-square factors of T
- **Centers**(U^2, T): centers of occurrences of U^2 in T

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then $C \in \textbf{PrimCov}[i]$ iff $C \in \textbf{Seeds}(T)$, $C^2 \in \textbf{PSquares}(T)$ and $n + i \in \textbf{Centers}(C^2, T)$.

Seeds in [CIRRSWZ, TCS 21]

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then C is a cover of $rot_i(S)$ iff C is a seed of T and C^2 occurs in T with center at position $n + i$.

String C is primitive if C is not a (full) power of a shorter string.

Observation. The shortest cover of a string is primitive.

U^2 is a p-square (primitively-rooted square) if U is primitive.

- **PrimCov**$[i]$: primitive covers of $rot_i(S)$
- **Seeds**(T): seeds of T
- **PSquares**(T): p-square factors of T
- **Centers**(U^2, T): centers of occurrences of U^2 in T

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then $C \in \textbf{PrimCov}[i]$ iff

$C \in \textbf{Seeds}(T)$, $C^2 \in \textbf{PSquares}(T)$ and $n + i \in \textbf{Centers}(C^2, T)$.

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then $C \in \textbf{PrimCov}[i]$ iff

$C \in \textbf{Seeds}(T)$, $C^2 \in \textbf{PSquares}(T)$ and $n + i \in \textbf{Centers}(C^2, T)$.

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in$ **PrimCov**$[i]$ iff

$C \in$ **Seeds**(T), $C^2 \in$ **PSquares**(T) and $n + i \in$ **Centers**(C^2, T).

`b a a b a a b a | b a a b a a b a | b a a b a a b a`

Seeds(T) of length $\le n$:

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

a b a a b a a b a | b a a b a a b a | b a a b a a b a

Seeds(T) of length $\leq n$: aba

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff $C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

`a b a a b a a b a | b a a b a a b a | b a a b a a b a a b`

Seeds(T) of length $\leq n$: `aba`, `abaab`

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in \textbf{PrimCov}[i]$ iff
$C \in \textbf{Seeds}(T)$, $C^2 \in \textbf{PSquares}(T)$ and $n + i \in \textbf{Centers}(C^2, T)$.

Seeds(T) of length $\le n$: aba, abaab, abaaba

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

`b a a b a a b a | b a a b a a b a | b a a b a a b a`

Seeds(T) of length $\le n$: `aba`, `abaab`, `abaaba`, `baaba`

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff $C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

b a a b a a b a | b a a b a a b a | b a a b a a b a

Seeds(T) of length $\le n$: aba, abaab, abaaba, baaba, $rot_i(S)$

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

NO

b a a b a a b a | b a a b a a b a | b a a b a a b a

Seeds(T) of length $\leq n$: aba, abaab, abaaba, baaba, $rot_i(S)$

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

CC **5**

Seeds(T) of length $\le n$: aba, abaab, abaaba, baaba, $rot_i(S)$

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

YES

b a a b a a b a | b a a b a a b a | b a a b a a b a

CC **5 8**

Seeds(T) of length $\le n$: aba, abaab, abaaba, baaba, $\underline{rot_i(S)}$

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in \textbf{PrimCov}[i]$ iff $C \in \textbf{Seeds}(T)$, $C^2 \in \textbf{PSquares}(T)$ and $n + i \in \textbf{Centers}(C^2, T)$.

NO

b a a b a a b a | b a a b a a b a | b a a b a a b a

CC **5 8**

Seeds(T) of length $\le n$: aba, abaab, abaaba, baaba, $rot_i(S)$

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff
$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

YES

b a a b a a b a | b a a b a a b a | b a a b a a b a

CC **5 8 3**

Seeds(T) of length $\le n$: <u>aba</u>, abaab, abaaba, baaba, $rot_i(S)$

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

NO

b a a b a a b a | b a a b a a b a | b a a b a a b a

CC **5 8 3**

Seeds(T) of length $\leq n$: aba, abaab, abaaba, baaba, $rot_i(S)$

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

YES

b a a b a a b a | b a a b a a b a | b a a b a a b a

CC **5 8 3 8**

Seeds(T) of length $\le n$: aba, abaab, abaaba, baaba, $\underline{rot_i(S)}$

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff $C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

for each $C^2 \in \mathbf{PSquares}(T)$
    if $C \in \mathbf{Seeds}(T)$ **then**
        for each $n + i \in \mathbf{Centers}(C^2, T)$
            if $i \in [0, n-1]$
                $\mathbf{PrimCov}[i] := \mathbf{PrimCov}[i] \cup \{C\}$

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff $C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

for each $C^2 \in \mathbf{PSquares}(T)$ // $\mathcal{O}(n)$ p-squares (1) and $\mathcal{O}(n)$ time (2)

  if $C \in \mathbf{Seeds}(T)$ **then**

    for each $n + i \in \mathbf{Centers}(C^2, T)$

      if $i \in [0, n-1]$

        $\mathbf{PrimCov}[i] := \mathbf{PrimCov}[i] \cup \{C\}$

(1) [Fraenkel and Simpson, JCTA 98], [Brlek and Li, 22]

(2) Crochemore+ [CIKRRW, TCS 14]

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

for each $C^2 \in \mathbf{PSquares}(T)$ // $\mathcal{O}(n)$ p-squares (1) and $\mathcal{O}(n)$ time (2)

  if $C \in \mathbf{Seeds}(T)$ **then** // $\mathcal{O}(1)$-time test after $\mathcal{O}(n)$ preprocessing (3)

    for each $n + i \in \mathbf{Centers}(C^2, T)$

      if $i \in [0, n-1]$

        $\mathbf{PrimCov}[i] := \mathbf{PrimCov}[i] \cup \{C\}$

(1) [Fraenkel and Simpson, JCTA 98], [Brlek and Li, 22]

(2) Crochemore+ [CIKRRW, TCS 14]

(3) Kociumaka+ [KKRRW, TALG 20], Belazzougui+ [BKPR, CPM 21]

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

for each $C^2 \in \mathbf{PSquares}(T)$ // $\mathcal{O}(n)$ p-squares (1) and $\mathcal{O}(n)$ time (2)

  if $C \in \mathbf{Seeds}(T)$ **then** // $\mathcal{O}(1)$-time test after $\mathcal{O}(n)$ preprocessing (3)

    for each $n + i \in \mathbf{Centers}(C^2, T)$ // $\mathcal{O}(n \log n)$ p-square occs (4)

      if $i \in [0, n-1]$

        $\mathbf{PrimCov}[i] := \mathbf{PrimCov}[i] \cup \{C\}$

(1) [Fraenkel and Simpson, JCTA 98], [Brlek and Li, 22]

(2) Crochemore+ [CIKRRW, TCS 14]

(3) Kociumaka+ [KKRRW, TALG 20], Belazzougui+ [BKPR, CPM 21]

(4) [Crochemore and Rytter, Algorithmica 95] and use suffix tree

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff $C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

for each $C^2 \in \mathbf{PSquares}(T)$ // $\mathcal{O}(n)$ p-squares (1) and $\mathcal{O}(n)$ time (2)

  if $C \in \mathbf{Seeds}(T)$ **then** // $\mathcal{O}(1)$-time test after $\mathcal{O}(n)$ preprocessing (3)

    for each $n + i \in \mathbf{Centers}(C^2, T)$

      if $i \in [0, n-1]$

        $\mathbf{PrimCov}[i] := \mathbf{PrimCov}[i] \cup \{C\}$

(1) [Fraenkel and Simpson, JCTA 98], [Brlek and Li, 22]

(2) Crochemore+ [CIKRRW, TCS 14]

(3) Kociumaka+ [KKRRW, TALG 20], Belazzougui+ [BKPR, CPM 21]

Observation. Running time is $\mathcal{O}(n + \sum_i |\mathbf{PrimCov}[i]|)$.

Algorithm from [CIRRSWZ, TCS 21] (Reformulated)

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

for each $C^2 \in \mathbf{PSquares}(T)$ // $\mathcal{O}(n)$ p-squares (1) and $\mathcal{O}(n)$ time (2)

  if $C \in \mathbf{Seeds}(T)$ **then** // $\mathcal{O}(1)$-time test after $\mathcal{O}(n)$ preprocessing (3)

    for each $n + i \in \mathbf{Centers}(C^2, T)$

      if $i \in [0, n-1]$

        $\mathbf{PrimCov}[i] := \mathbf{PrimCov}[i] \cup \{C\}$

(1) [Fraenkel and Simpson, JCTA 98], [Brlek and Li, 22]

(2) Crochemore+ [CIKRRW, TCS 14]

(3) Kociumaka+ [KKRRW, TALG 20], Belazzougui+ [BKPR, CPM 21]

Observation. Running time is $\mathcal{O}(n + \sum_i |\mathbf{PrimCov}[i]|)$.

Theorem (This paper). $\sum_i |\mathbf{PrimCov}[i]|$ can be $\Theta(n \log n)$. ☹

Our Idea 1. Application of Runs

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

b a a b a a b a | b a a b a a b a | b a a b a a b a

Our Idea 1. Application of Runs

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then $C \in \textbf{PrimCov}[i]$ iff

$C \in \textbf{Seeds}(T)$, $C^2 \in \textbf{PSquares}(T)$ and $n + i \in \textbf{Centers}(C^2, T)$.

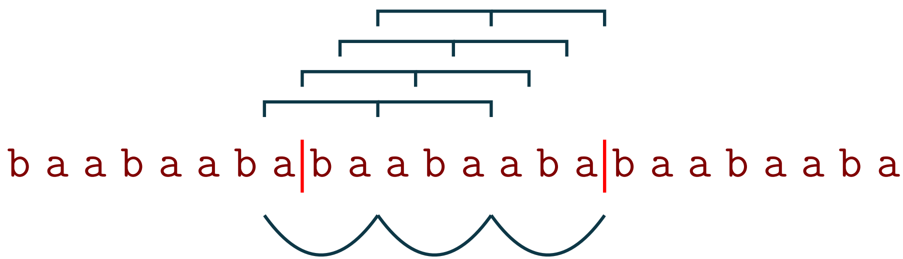
b a a b a a b a | b a a b a a b a | b a a b a a b a

Our Idea 1. Application of Runs

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in \textbf{PrimCov}[i]$ iff

$C \in \textbf{Seeds}(T)$, $C^2 \in \textbf{PSquares}(T)$ and $n + i \in \textbf{Centers}(C^2, T)$.



Definition (run)

A run is an inclusion-maximal periodic fragment.

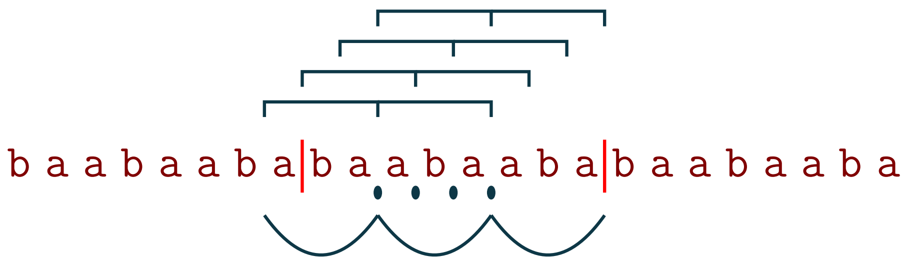
(String U is periodic if its smallest period p satisfies $p \le |U|/2$.)

Our Idea 1. Application of Runs

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.



Definition (run)

A run is an inclusion-maximal periodic fragment.

(String U is periodic if its smallest period p satisfies $p \le |U|/2$.)

Observation

Run $T[i..j]$ with period $p \Rightarrow$ p-squares with centers in $[i + p..j - p + 1]$.

Our Idea 1. Application of Runs

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

Observation

Run $T[i..j]$ with period $p \Rightarrow$ p-squares with centers in $[i + p..j - p + 1]$.

Our Idea 1. Application of Runs

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then $C \in \textbf{PrimCov}[i]$ iff

$C \in \textbf{Seeds}(T)$, $C^2 \in \textbf{PSquares}(T)$ and $n + i \in \textbf{Centers}(C^2, T)$.

Observation

Run $T[i..j]$ with period $p \Rightarrow$ p-squares with centers in $[i + p..j - p + 1]$.

Properties of runs:

- $\mathcal{O}(n)$ runs in T [Kolpakov and Kucherov, FOCS 99]
- runs computed in $\mathcal{O}(n)$ time [Kolpakov and Kucherov, FOCS 99]

  also over any ordered alphabet [Ellert and Fischer, ICALP 21]

Our Idea 1. Application of Runs

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

Observation

Run $T[i..j]$ with period $p \Rightarrow$ p-squares with centers in $[i + p..j - p + 1]$.

Properties of runs:

- $\mathcal{O}(n)$ runs in T [Kolpakov and Kucherov, FOCS 99]
- runs computed in $\mathcal{O}(n)$ time [Kolpakov and Kucherov, FOCS 99]

  also over any ordered alphabet [Ellert and Fischer, ICALP 21]

Our Idea 2. Global Seed Mask for p-Squares

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

Observation

Run $T[i..j]$ with period $p \Rightarrow$ p-squares with centers in $[i + p..j - p + 1]$.

p-Squares in some string T:

- $\mathtt{a}^2$
- $(\mathtt{ab})^2$
- $(\mathtt{ba})^2$
- $(\mathtt{aba})^2$
- $(\mathtt{aaab})^2$
- $(\mathtt{aaba})^2$
- $(\mathtt{baaa})^2$
- $(\mathtt{aaaba})^2$
- $(\mathtt{aabaa})^2$
- $(\mathtt{aabab})^2$
- $(\mathtt{abaaa})^2$
- $(\mathtt{ababa})^2$
- $(\mathtt{aaabab})^2$
- $(\mathtt{abaaaa})^2$

Our Idea 2. Global Seed Mask for p-Squares

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

Observation

Run $T[i..j]$ with period $p \Rightarrow$ p-squares with centers in $[i + p..j - p + 1]$.

p-Squares in some string T, half is seed or no seed:

- $\mathtt{a}^2$ NO
- $(\mathtt{ab})^2$ NO
- $(\mathtt{ba})^2$ NO
- $(\mathtt{aba})^2$ NO
- $(\mathtt{aaab})^2$ NO
- $(\mathtt{aaba})^2$ YES
- $(\mathtt{baaa})^2$ NO
- $(\mathtt{aaaba})^2$ NO
- $(\mathtt{aabaa})^2$ YES
- $(\mathtt{aabab})^2$ NO
- $(\mathtt{abaaa})^2$ YES
- $(\mathtt{ababa})^2$ NO
- $(\mathtt{aaabab})^2$ YES
- $(\mathtt{abaaaa})^2$ NO

Our Idea 2. Global Seed Mask for p-Squares

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

Observation

Run $T[i..j]$ with period $p \Rightarrow$ p-squares with centers in $[i+p..j-p+1]$.

p-Squares in some string T, half is seed or no seed, Lyndon root:

- $\mathtt{a}^2$ NO → a
- $(\mathtt{ab})^2$ NO → ab
- $(\mathtt{ba})^2$ NO → ab
- $(\mathtt{aba})^2$ NO → aab
- $(\mathtt{aaab})^2$ NO → aaab
- $(\mathtt{aaba})^2$ YES → aaab
- $(\mathtt{baaa})^2$ NO → aaab
- $(\mathtt{aaaba})^2$ NO → aaaab
- $(\mathtt{aabaa})^2$ YES → aaaab
- $(\mathtt{aabab})^2$ NO → aabab
- $(\mathtt{abaaa})^2$ YES → aaaab
- $(\mathtt{ababa})^2$ NO → aabab
- $(\mathtt{aaabab})^2$ YES → aaabab
- $(\mathtt{abaaaa})^2$ NO → aaaaab

Our Idea 2. Global Seed Mask for p-Squares

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

Observation

Run $T[i..j]$ with period $p \Rightarrow$ p-squares with centers in $[i + p..j - p + 1]$.

p-Squares in some string T, half is seed or no seed, rotation of Lyndon root:

- $\mathtt{a}^2$ NO $\rightarrow rot_0(\mathtt{a})^2$
- $(\mathtt{ab})^2$ NO $\rightarrow rot_0(\mathtt{ab})^2$
- $(\mathtt{ba})^2$ NO $\rightarrow rot_1(\mathtt{ab})^2$
- $(\mathtt{aba})^2$ NO $\rightarrow rot_1(\mathtt{aab})^2$
- $(\mathtt{aaab})^2$ NO $\rightarrow rot_0(\mathtt{aaab})^2$
- $(\mathtt{aaba})^2$ YES $\rightarrow rot_1(\mathtt{aaab})^2$
- $(\mathtt{baaa})^2$ NO $\rightarrow rot_3(\mathtt{aaab})^2$
- $(\mathtt{aaaba})^2$ NO $\rightarrow rot_1(\mathtt{aaaab})^2$
- $(\mathtt{aabaa})^2$ YES $\rightarrow rot_2(\mathtt{aaaab})^2$
- $(\mathtt{aabab})^2$ NO $\rightarrow rot_0(\mathtt{aabab})^2$
- $(\mathtt{abaaa})^2$ YES $\rightarrow rot_3(\mathtt{aaaab})^2$
- $(\mathtt{ababa})^2$ NO $\rightarrow rot_1(\mathtt{aabab})^2$
- $(\mathtt{aaabab})^2$ YES $\rightarrow rot_0(\mathtt{aaabab})^2$
- $(\mathtt{abaaaa})^2$ NO $\rightarrow rot_4(\mathtt{aaaaab})^2$

Our Idea 2. Global Seed Mask for p-Squares

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \leq |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

Observation

Run $T[i..j]$ with period $p \Rightarrow$ p-squares with centers in $[i + p..j - p + 1]$.

p-Squares in some string T, half is seed or no seed, rotation of Lyndon root:

- $\mathtt{a}^2$ NO $\rightarrow rot_0(\mathtt{a})^2$
- $(\mathtt{ab})^2$ NO $\rightarrow rot_0(\mathtt{ab})^2$
- $(\mathtt{ba})^2$ NO $\rightarrow rot_1(\mathtt{ab})^2$
- $(\mathtt{aba})^2$ NO $\rightarrow rot_1(\mathtt{aab})^2$
- $(\mathtt{aaab})^2$ NO $\rightarrow rot_0(\mathtt{aaab})^2$
- $(\mathtt{aaba})^2$ YES $\rightarrow rot_1(\mathtt{aaab})^2$
- $(\mathtt{baaa})^2$ NO $\rightarrow rot_3(\mathtt{aaab})^2$
- $(\mathtt{aaaba})^2$ NO $\rightarrow rot_1(\mathtt{aaaab})^2$
- $(\mathtt{aabaa})^2$ YES $\rightarrow rot_2(\mathtt{aaaab})^2$
- $(\mathtt{abaaa})^2$ YES $\rightarrow rot_3(\mathtt{aaaab})^2$
- $(\mathtt{aabab})^2$ NO $\rightarrow rot_0(\mathtt{aabab})^2$
- $(\mathtt{ababa})^2$ NO $\rightarrow rot_1(\mathtt{aabab})^2$
- $(\mathtt{abaaaa})^2$ NO $\rightarrow rot_4(\mathtt{aaaaab})^2$
- $(\mathtt{aaabab})^2$ YES $\rightarrow rot_0(\mathtt{aaabab})^2$

Our Idea 2. Global Seed Mask for p-Squares

Lemma (Covers of Rotations = Seeds + Squares, [CIRRSWZ, TCS 21])

If $|C| \le |S|$ and $T = S^3$, then $C \in \mathbf{PrimCov}[i]$ iff

$C \in \mathbf{Seeds}(T)$, $C^2 \in \mathbf{PSquares}(T)$ and $n + i \in \mathbf{Centers}(C^2, T)$.

Observation

Run $T[i..j]$ with period $p \Rightarrow$ p-squares with centers in $[i + p..j - p + 1]$.

p-Squares in some string T, half is seed or no seed, rotation of Lyndon root:

- $\mathtt{a}^2$ NO $\to rot_0(\mathtt{a})^2$
- $(\mathtt{ab})^2$ NO $\to rot_0(\mathtt{ab})^2$
- $(\mathtt{ba})^2$ NO $\to rot_1(\mathtt{ab})^2$
- $(\mathtt{aba})^2$ NO $\to rot_1(\mathtt{aab})^2$
- $(\mathtt{aaab})^2$ NO $\to rot_0(\mathtt{aaab})^2$
- $(\mathtt{aaba})^2$ YES $\to rot_1(\mathtt{aaab})^2$
- $(\mathtt{baaa})^2$ NO $\to rot_3(\mathtt{aaab})^2$
- $(\mathtt{aaaba})^2$ NO $\to rot_1(\mathtt{aaaab})^2$
- $(\mathtt{aabaa})^2$ YES $\to rot_2(\mathtt{aaaab})^2$
- $(\mathtt{abaaa})^2$ YES $\to rot_3(\mathtt{aaaab})^2$
- $(\mathtt{aabab})^2$ NO $\to rot_0(\mathtt{aabab})^2$
- $(\mathtt{ababa})^2$ NO $\to rot_1(\mathtt{aabab})^2$
- $(\mathtt{abaaaa})^2$ NO $\to rot_4(\mathtt{aaaaab})^2$
- $(\mathtt{aaabab})^2$ YES $\to rot_0(\mathtt{aaabab})^2$

Seed mask: 0 0 0 0 0 1 0 0 1 1 0 0 0 1

Our Idea 2. Global Seed Mask for p-Squares

p-Squares in some string T, half is seed or no seed, rotation of Lyndon root:

- $\mathtt{a}^2$ NO → $rot_0(\mathtt{a})^2$
- $(\mathtt{ab})^2$ NO → $rot_0(\mathtt{ab})^2$
- $(\mathtt{ba})^2$ NO → $rot_1(\mathtt{ab})^2$
- $(\mathtt{aba})^2$ NO → $rot_1(\mathtt{aab})^2$
- $(\mathtt{aaab})^2$ NO → $rot_0(\mathtt{aaab})^2$
- $(\mathtt{aaba})^2$ YES → $rot_1(\mathtt{aaab})^2$
- $(\mathtt{baaa})^2$ NO → $rot_3(\mathtt{aaab})^2$
- $(\mathtt{aaaba})^2$ NO → $rot_1(\mathtt{aaaab})^2$
- $(\mathtt{aabaa})^2$ YES → $rot_2(\mathtt{aaaab})^2$
- $(\mathtt{abaaa})^2$ YES → $rot_3(\mathtt{aaaab})^2$
- $(\mathtt{aabab})^2$ NO → $rot_0(\mathtt{aabab})^2$
- $(\mathtt{ababa})^2$ NO → $rot_1(\mathtt{aabab})^2$
- $(\mathtt{abaaaa})^2$ NO → $rot_4(\mathtt{aaaaab})^2$
- $(\mathtt{aaabab})^2$ YES → $rot_0(\mathtt{aaabab})^2$

Seed mask: 0 0 0 0 0 1 0 0 1 1 0 0 0 1

Our Idea 2. Global Seed Mask for p-Squares

p-Squares in some string T, half is seed or no seed, rotation of Lyndon root:

- $\mathtt{a}^2$ NO $\rightarrow rot_0(\mathtt{a})^2$
- $(\mathtt{ab})^2$ NO $\rightarrow rot_0(\mathtt{ab})^2$
- $(\mathtt{ba})^2$ NO $\rightarrow rot_1(\mathtt{ab})^2$
- $(\mathtt{aba})^2$ NO $\rightarrow rot_1(\mathtt{aab})^2$
- $(\mathtt{aaab})^2$ NO $\rightarrow rot_0(\mathtt{aaab})^2$
- $(\mathtt{aaba})^2$ YES $\rightarrow rot_1(\mathtt{aaab})^2$
- $(\mathtt{baaa})^2$ NO $\rightarrow rot_3(\mathtt{aaab})^2$
- $(\mathtt{aaaba})^2$ NO $\rightarrow rot_1(\mathtt{aaaab})^2$
- $(\mathtt{aabaa})^2$ YES $\rightarrow rot_2(\mathtt{aaaab})^2$
- $(\mathtt{abaaa})^2$ YES $\rightarrow rot_3(\mathtt{aaaab})^2$
- $(\mathtt{aabab})^2$ NO $\rightarrow rot_0(\mathtt{aabab})^2$
- $(\mathtt{ababa})^2$ NO $\rightarrow rot_1(\mathtt{aabab})^2$
- $(\mathtt{abaaaa})^2$ NO $\rightarrow rot_4(\mathtt{aaaaab})^2$
- $(\mathtt{aaabab})^2$ YES $\rightarrow rot_0(\mathtt{aaabab})^2$

Seed mask: 0 0 0 0 0 1 0 0 1 1 0 0 0 1

Our Idea 2. Global Seed Mask for p-Squares

p-Squares in some string T, half is seed or no seed, rotation of Lyndon root:

- $\mathtt{a}^2$ NO → $rot_0(\mathtt{a})^2$
- $(\mathtt{ab})^2$ NO → $rot_0(\mathtt{ab})^2$
- $(\mathtt{ba})^2$ NO → $rot_1(\mathtt{ab})^2$
- $(\mathtt{aba})^2$ NO → $rot_1(\mathtt{aab})^2$
- $(\mathtt{aaab})^2$ NO → $rot_0(\mathtt{aaab})^2$
- $(\mathtt{aaba})^2$ YES → $rot_1(\mathtt{aaab})^2$
- $(\mathtt{baaa})^2$ NO → $rot_3(\mathtt{aaab})^2$
- $(\mathtt{aaaba})^2$ NO → $rot_1(\mathtt{aaaab})^2$
- $(\mathtt{aabaa})^2$ YES → $rot_2(\mathtt{aaaab})^2$
- $(\mathtt{abaaa})^2$ YES → $rot_3(\mathtt{aaaab})^2$
- $(\mathtt{aabab})^2$ NO → $rot_0(\mathtt{aabab})^2$
- $(\mathtt{ababa})^2$ NO → $rot_1(\mathtt{aabab})^2$
- $(\mathtt{abaaaa})^2$ NO → $rot_4(\mathtt{aaaaab})^2$
- $(\mathtt{aaabab})^2$ YES → $rot_0(\mathtt{aaabab})^2$

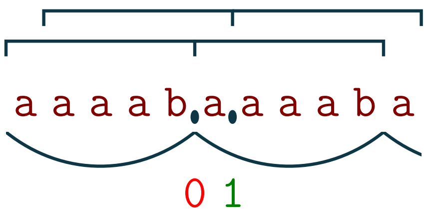
Seed mask: 0 0 0 0 0 1 0 0 1 1 0 0 0 1

Our Idea 2. Global Seed Mask for p-Squares

p-Squares in some string T, half is seed or no seed, rotation of Lyndon root:

- $\mathtt{a}^2$ NO $\rightarrow rot_0(\mathtt{a})^2$
- $(\mathtt{ab})^2$ NO $\rightarrow rot_0(\mathtt{ab})^2$
- $(\mathtt{ba})^2$ NO $\rightarrow rot_1(\mathtt{ab})^2$
- $(\mathtt{aba})^2$ NO $\rightarrow rot_1(\mathtt{aab})^2$
- $(\mathtt{aaab})^2$ NO $\rightarrow rot_0(\mathtt{aaab})^2$
- $(\mathtt{aaba})^2$ YES $\rightarrow rot_1(\mathtt{aaab})^2$
- $(\mathtt{baaa})^2$ NO $\rightarrow rot_3(\mathtt{aaab})^2$
- $(\mathtt{aaaba})^2$ NO $\rightarrow rot_1(\mathtt{aaaab})^2$
- $(\mathtt{aabaa})^2$ YES $\rightarrow rot_2(\mathtt{aaaab})^2$
- $(\mathtt{abaaa})^2$ YES $\rightarrow rot_3(\mathtt{aaaab})^2$
- $(\mathtt{aabab})^2$ NO $\rightarrow rot_0(\mathtt{aabab})^2$
- $(\mathtt{ababa})^2$ NO $\rightarrow rot_1(\mathtt{aabab})^2$
- $(\mathtt{abaaaa})^2$ NO $\rightarrow rot_4(\mathtt{aaaaab})^2$
- $(\mathtt{aaabab})^2$ YES $\rightarrow rot_0(\mathtt{aaabab})^2$

Seed mask: 0 0 0 0 0 1 0 0 1 1 0 0 0 1



Our Idea 2. Global Seed Mask for p-Squares

p-Squares in some string T, half is seed or no seed, rotation of Lyndon root:

- $\texttt{a}^2$ NO $\rightarrow rot_0(\texttt{a})^2$
- $(\texttt{ab})^2$ NO $\rightarrow rot_0(\texttt{ab})^2$
- $(\texttt{ba})^2$ NO $\rightarrow rot_1(\texttt{ab})^2$
- $(\texttt{aba})^2$ NO $\rightarrow rot_1(\texttt{aab})^2$
- $(\texttt{aaab})^2$ NO $\rightarrow rot_0(\texttt{aaab})^2$
- $(\texttt{aaba})^2$ YES $\rightarrow rot_1(\texttt{aaab})^2$
- $(\texttt{baaa})^2$ NO $\rightarrow rot_3(\texttt{aaab})^2$
- $(\texttt{aaaba})^2$ NO $\rightarrow rot_1(\texttt{aaaab})^2$
- $(\texttt{aabaa})^2$ YES $\rightarrow rot_2(\texttt{aaaab})^2$
- $(\texttt{abaaa})^2$ YES $\rightarrow rot_3(\texttt{aaaab})^2$
- $(\texttt{aabab})^2$ NO $\rightarrow rot_0(\texttt{aabab})^2$
- $(\texttt{ababa})^2$ NO $\rightarrow rot_1(\texttt{aabab})^2$
- $(\texttt{abaaaa})^2$ NO $\rightarrow rot_4(\texttt{aaaaab})^2$
- $(\texttt{aaabab})^2$ YES $\rightarrow rot_0(\texttt{aaabab})^2$

Seed mask: 0 0 0 0 0 1 0 0 1 1 0 0 0 1

Our Idea 2. Global Seed Mask for p-Squares

p-Squares in some string T, half is seed or no seed, rotation of Lyndon root:

- $\mathtt{a}^2$ NO $\rightarrow$ $rot_0(\mathtt{a})^2$
- $(\mathtt{ab})^2$ NO $\rightarrow$ $rot_0(\mathtt{ab})^2$
- $(\mathtt{ba})^2$ NO $\rightarrow$ $rot_1(\mathtt{ab})^2$
- $(\mathtt{aba})^2$ NO $\rightarrow$ $rot_1(\mathtt{aab})^2$
- $(\mathtt{aaab})^2$ NO $\rightarrow$ $rot_0(\mathtt{aaab})^2$
- $(\mathtt{aaba})^2$ YES $\rightarrow$ $rot_1(\mathtt{aaab})^2$
- $(\mathtt{baaa})^2$ NO $\rightarrow$ $rot_3(\mathtt{aaab})^2$
- $(\mathtt{aaaba})^2$ NO $\rightarrow$ $rot_1(\mathtt{aaaab})^2$
- $(\mathtt{aabaa})^2$ YES $\rightarrow$ $rot_2(\mathtt{aaaab})^2$
- $(\mathtt{abaaa})^2$ YES $\rightarrow$ $rot_3(\mathtt{aaaab})^2$
- $(\mathtt{aabab})^2$ NO $\rightarrow$ $rot_0(\mathtt{aabab})^2$
- $(\mathtt{ababa})^2$ NO $\rightarrow$ $rot_1(\mathtt{aabab})^2$
- $(\mathtt{abaaaa})^2$ NO $\rightarrow$ $rot_4(\mathtt{aaaaab})^2$
- $(\mathtt{aaabab})^2$ YES $\rightarrow$ $rot_0(\mathtt{aaabab})^2$

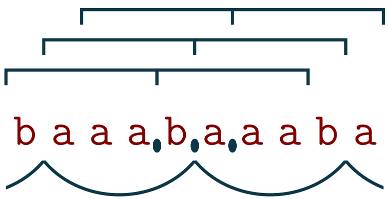
Seed mask: 0 0 0 0 0 1 0 0 1 1 0 0 0 1

Our Idea 2. Global Seed Mask for p-Squares

p-Squares in some string T, half is seed or no seed, rotation of Lyndon root:

- a^2 NO → $rot_0(a)^2$
- $(ab)^2$ NO → $rot_0(ab)^2$
- $(ba)^2$ NO → $rot_1(ab)^2$
- $(aba)^2$ NO → $rot_1(aab)^2$
- $(aaab)^2$ NO → $rot_0(aaab)^2$
- $(aaba)^2$ YES → $rot_1(aaab)^2$
- $(baaa)^2$ NO → $rot_3(aaab)^2$
- $(aaaba)^2$ NO → $rot_1(aaaab)^2$
- $(aabaa)^2$ YES → $rot_2(aaaab)^2$
- $(abaaa)^2$ YES → $rot_3(aaaab)^2$
- $(aabab)^2$ NO → $rot_0(aabab)^2$
- $(ababa)^2$ NO → $rot_1(aabab)^2$
- $(abaaaa)^2$ NO → $rot_4(aaaaab)^2$
- $(aaabab)^2$ YES → $rot_0(aaabab)^2$

Seed mask: 0 0 0 0 0 1 0 0 1 1 0 0 0 1

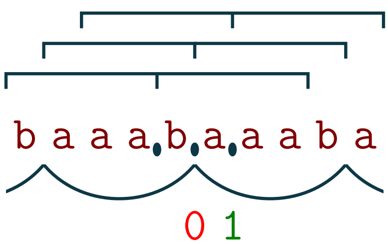


Our Idea 2. Global Seed Mask for p-Squares

p-Squares in some string T, half is seed or no seed, rotation of Lyndon root:

- $\mathtt{a}^2$ NO $\rightarrow rot_0(\mathtt{a})^2$
- $(\mathtt{ab})^2$ NO $\rightarrow rot_0(\mathtt{ab})^2$
- $(\mathtt{ba})^2$ NO $\rightarrow rot_1(\mathtt{ab})^2$
- $(\mathtt{aba})^2$ NO $\rightarrow rot_1(\mathtt{aab})^2$
- $(\mathtt{aaab})^2$ NO $\rightarrow rot_0(\mathtt{aaab})^2$
- $(\mathtt{aaba})^2$ YES $\rightarrow rot_1(\mathtt{aaab})^2$
- $(\mathtt{baaa})^2$ NO $\rightarrow rot_3(\mathtt{aaab})^2$
- $(\mathtt{aaaba})^2$ NO $\rightarrow rot_1(\mathtt{aaaab})^2$
- $(\mathtt{aabaa})^2$ YES $\rightarrow rot_2(\mathtt{aaaab})^2$
- $(\mathtt{abaaa})^2$ YES $\rightarrow rot_3(\mathtt{aaaab})^2$
- $(\mathtt{aabab})^2$ NO $\rightarrow rot_0(\mathtt{aabab})^2$
- $(\mathtt{ababa})^2$ NO $\rightarrow rot_1(\mathtt{aabab})^2$
- $(\mathtt{abaaaa})^2$ NO $\rightarrow rot_4(\mathtt{aaaaab})^2$
- $(\mathtt{aaabab})^2$ YES $\rightarrow rot_0(\mathtt{aaabab})^2$

Seed mask: 0 0 0 0 0 1 0 0 1 1 0 0 0 1

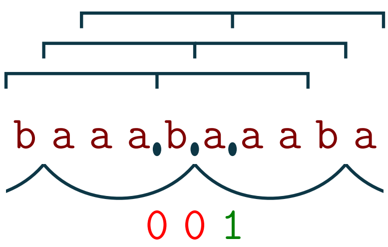


Our Idea 2. Global Seed Mask for p-Squares

p-Squares in some string T, half is seed or no seed, rotation of Lyndon root:

- $\mathtt{a}^2$ NO $\rightarrow rot_0(\mathtt{a})^2$
- $(\mathtt{ab})^2$ NO $\rightarrow rot_0(\mathtt{ab})^2$
- $(\mathtt{ba})^2$ NO $\rightarrow rot_1(\mathtt{ab})^2$
- $(\mathtt{aba})^2$ NO $\rightarrow rot_1(\mathtt{aab})^2$
- $(\mathtt{aaab})^2$ NO $\rightarrow rot_0(\mathtt{aaab})^2$
- $(\mathtt{aaba})^2$ YES $\rightarrow rot_1(\mathtt{aaab})^2$
- $(\mathtt{baaa})^2$ NO $\rightarrow rot_3(\mathtt{aaab})^2$
- $(\mathtt{aaaba})^2$ NO $\rightarrow rot_1(\mathtt{aaaab})^2$
- $(\mathtt{aabaa})^2$ YES $\rightarrow rot_2(\mathtt{aaaab})^2$
- $(\mathtt{abaaa})^2$ YES $\rightarrow rot_3(\mathtt{aaaab})^2$
- $(\mathtt{aabab})^2$ NO $\rightarrow rot_0(\mathtt{aabab})^2$
- $(\mathtt{ababa})^2$ NO $\rightarrow rot_1(\mathtt{aabab})^2$
- $(\mathtt{abaaaa})^2$ NO $\rightarrow rot_4(\mathtt{aaaaab})^2$
- $(\mathtt{aaabab})^2$ YES $\rightarrow rot_0(\mathtt{aaabab})^2$

Seed mask: 0 0 0 0 0 1 0 0 1 1 0 0 0 1



Our Idea 2. Global Seed Mask for p-Squares

p-Squares in some string T, half is seed or no seed, rotation of Lyndon root:

- a^2 NO → $rot_0(a)^2$
- $(ab)^2$ NO → $rot_0(ab)^2$
- $(ba)^2$ NO → $rot_1(ab)^2$
- $(aba)^2$ NO → $rot_1(aab)^2$
- $(aaab)^2$ NO → $rot_0(aaab)^2$
- $(aaba)^2$ YES → $rot_1(aaab)^2$
- $(baaa)^2$ NO → $rot_3(aaab)^2$
- $(aaaba)^2$ NO → $rot_1(aaaab)^2$
- $(aabaa)^2$ YES → $rot_2(aaaab)^2$
- $(abaaa)^2$ YES → $rot_3(aaaab)^2$
- $(aabab)^2$ NO → $rot_0(aabab)^2$
- $(ababa)^2$ NO → $rot_1(aabab)^2$
- $(abaaaa)^2$ NO → $rot_4(aaaaab)^2$
- $(aaabab)^2$ YES → $rot_0(aaabab)^2$

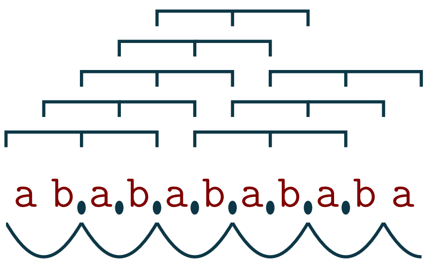
Seed mask: 0 0 0 0 0 1 0 0 1 1 0 0 0 1

Our Idea 2. Global Seed Mask for p-Squares

p-Squares in some string T, half is seed or no seed, rotation of Lyndon root:

- $\mathtt{a}^2$ NO → $rot_0(\mathtt{a})^2$
- $(\mathtt{ab})^2$ NO → $rot_0(\mathtt{ab})^2$
- $(\mathtt{ba})^2$ NO → $rot_1(\mathtt{ab})^2$
- $(\mathtt{aba})^2$ NO → $rot_1(\mathtt{aab})^2$
- $(\mathtt{aaab})^2$ NO → $rot_0(\mathtt{aaab})^2$
- $(\mathtt{aaba})^2$ YES → $rot_1(\mathtt{aaab})^2$
- $(\mathtt{baaa})^2$ NO → $rot_3(\mathtt{aaab})^2$
- $(\mathtt{aaaba})^2$ NO → $rot_1(\mathtt{aaaab})^2$
- $(\mathtt{aabaa})^2$ YES → $rot_2(\mathtt{aaaab})^2$
- $(\mathtt{abaaa})^2$ YES → $rot_3(\mathtt{aaaab})^2$
- $(\mathtt{aabab})^2$ NO → $rot_0(\mathtt{aabab})^2$
- $(\mathtt{ababa})^2$ NO → $rot_1(\mathtt{aabab})^2$
- $(\mathtt{abaaaa})^2$ NO → $rot_4(\mathtt{aaaaab})^2$
- $(\mathtt{aaabab})^2$ YES → $rot_0(\mathtt{aaabab})^2$

Seed mask: 0 0 0 0 0 1 0 0 1 1 0 0 0 1

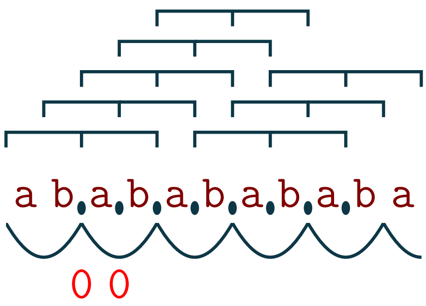


Our Idea 2. Global Seed Mask for p-Squares

p-Squares in some string T, half is seed or no seed, rotation of Lyndon root:

- $\mathtt{a}^2$ NO $\rightarrow rot_0(\mathtt{a})^2$
- $(\mathtt{ab})^2$ NO $\rightarrow rot_0(\mathtt{ab})^2$
- $(\mathtt{ba})^2$ NO $\rightarrow rot_1(\mathtt{ab})^2$
- $(\mathtt{aba})^2$ NO $\rightarrow rot_1(\mathtt{aab})^2$
- $(\mathtt{aaab})^2$ NO $\rightarrow rot_0(\mathtt{aaab})^2$
- $(\mathtt{aaba})^2$ YES $\rightarrow rot_1(\mathtt{aaab})^2$
- $(\mathtt{baaa})^2$ NO $\rightarrow rot_3(\mathtt{aaab})^2$
- $(\mathtt{aaaba})^2$ NO $\rightarrow rot_1(\mathtt{aaaab})^2$
- $(\mathtt{aabaa})^2$ YES $\rightarrow rot_2(\mathtt{aaaab})^2$
- $(\mathtt{abaaa})^2$ YES $\rightarrow rot_3(\mathtt{aaaab})^2$
- $(\mathtt{aabab})^2$ NO $\rightarrow rot_0(\mathtt{aabab})^2$
- $(\mathtt{ababa})^2$ NO $\rightarrow rot_1(\mathtt{aabab})^2$
- $(\mathtt{abaaaa})^2$ NO $\rightarrow rot_4(\mathtt{aaaaab})^2$
- $(\mathtt{aaabab})^2$ YES $\rightarrow rot_0(\mathtt{aaabab})^2$

Seed mask: 0 0 0 0 0 1 0 0 1 1 0 0 0 1

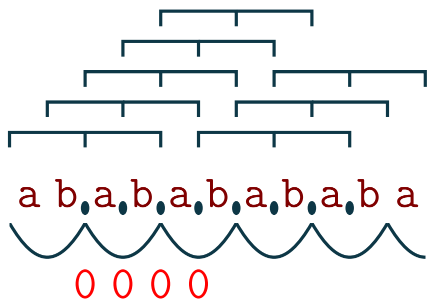


Our Idea 2. Global Seed Mask for p-Squares

p-Squares in some string T, half is seed or no seed, rotation of Lyndon root:

- $\mathtt{a}^2$ NO → $rot_0(\mathtt{a})^2$
- $(\mathtt{ab})^2$ NO → $rot_0(\mathtt{ab})^2$
- $(\mathtt{ba})^2$ NO → $rot_1(\mathtt{ab})^2$
- $(\mathtt{aba})^2$ NO → $rot_1(\mathtt{aab})^2$
- $(\mathtt{aaab})^2$ NO → $rot_0(\mathtt{aaab})^2$
- $(\mathtt{aaba})^2$ YES → $rot_1(\mathtt{aaab})^2$
- $(\mathtt{baaa})^2$ NO → $rot_3(\mathtt{aaab})^2$
- $(\mathtt{aaaba})^2$ NO → $rot_1(\mathtt{aaaab})^2$
- $(\mathtt{aabaa})^2$ YES → $rot_2(\mathtt{aaaab})^2$
- $(\mathtt{abaaa})^2$ YES → $rot_3(\mathtt{aaaab})^2$
- $(\mathtt{aabab})^2$ NO → $rot_0(\mathtt{aabab})^2$
- $(\mathtt{ababa})^2$ NO → $rot_1(\mathtt{aabab})^2$
- $(\mathtt{abaaaa})^2$ NO → $rot_4(\mathtt{aaaaab})^2$
- $(\mathtt{aaabab})^2$ YES → $rot_0(\mathtt{aaabab})^2$

Seed mask: 0 0 0 0 0 1 0 0 1 1 0 0 0 1

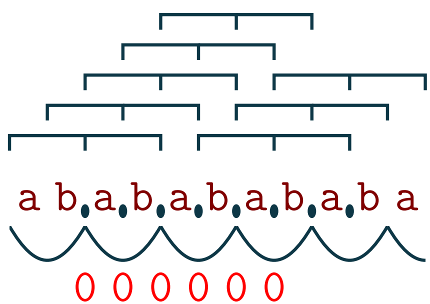


Our Idea 2. Global Seed Mask for p-Squares

p-Squares in some string T, half is seed or no seed, rotation of Lyndon root:

- $\mathtt{a}^2$ NO → $rot_0(\mathtt{a})^2$
- $(\mathtt{ab})^2$ NO → $rot_0(\mathtt{ab})^2$
- $(\mathtt{ba})^2$ NO → $rot_1(\mathtt{ab})^2$
- $(\mathtt{aba})^2$ NO → $rot_1(\mathtt{aab})^2$
- $(\mathtt{aaab})^2$ NO → $rot_0(\mathtt{aaab})^2$
- $(\mathtt{aaba})^2$ YES → $rot_1(\mathtt{aaab})^2$
- $(\mathtt{baaa})^2$ NO → $rot_3(\mathtt{aaab})^2$
- $(\mathtt{aaaba})^2$ NO → $rot_1(\mathtt{aaaab})^2$
- $(\mathtt{aabaa})^2$ YES → $rot_2(\mathtt{aaaab})^2$
- $(\mathtt{abaaa})^2$ YES → $rot_3(\mathtt{aaaab})^2$
- $(\mathtt{aabab})^2$ NO → $rot_0(\mathtt{aabab})^2$
- $(\mathtt{ababa})^2$ NO → $rot_1(\mathtt{aabab})^2$
- $(\mathtt{abaaaa})^2$ NO → $rot_4(\mathtt{aaaaab})^2$
- $(\mathtt{aaabab})^2$ YES → $rot_0(\mathtt{aaabab})^2$

Seed mask: 0 0 0 0 0 1 0 0 1 1 0 0 0 1

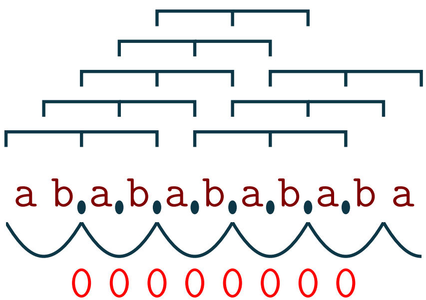


Our Idea 2. Global Seed Mask for p-Squares

p-Squares in some string T, half is seed or no seed, rotation of Lyndon root:

- $\mathtt{a}^2$ NO $\rightarrow rot_0(\mathtt{a})^2$
- $(\mathtt{ab})^2$ NO $\rightarrow rot_0(\mathtt{ab})^2$
- $(\mathtt{ba})^2$ NO $\rightarrow rot_1(\mathtt{ab})^2$
- $(\mathtt{aba})^2$ NO $\rightarrow rot_1(\mathtt{aab})^2$
- $(\mathtt{aaab})^2$ NO $\rightarrow rot_0(\mathtt{aaab})^2$
- $(\mathtt{aaba})^2$ YES $\rightarrow rot_1(\mathtt{aaab})^2$
- $(\mathtt{baaa})^2$ NO $\rightarrow rot_3(\mathtt{aaab})^2$
- $(\mathtt{aaaba})^2$ NO $\rightarrow rot_1(\mathtt{aaaab})^2$
- $(\mathtt{aabaa})^2$ YES $\rightarrow rot_2(\mathtt{aaaab})^2$
- $(\mathtt{abaaa})^2$ YES $\rightarrow rot_3(\mathtt{aaaab})^2$
- $(\mathtt{aabab})^2$ NO $\rightarrow rot_0(\mathtt{aabab})^2$
- $(\mathtt{ababa})^2$ NO $\rightarrow rot_1(\mathtt{aabab})^2$
- $(\mathtt{abaaaa})^2$ NO $\rightarrow rot_4(\mathtt{aaaaab})^2$
- $(\mathtt{aaabab})^2$ YES $\rightarrow rot_0(\mathtt{aaabab})^2$

Seed mask: 0 0 0 0 0 1 0 0 1 1 0 0 0 1



Our Idea 2. Global Seed Mask for p-Squares

Summary:

- Global seed mask of p-squares with $\mathcal{O}(n)$ bits
- Local seed mask for each run R with $\mathcal{O}(exp(R))$ chunks
  ($exp(R) = |R|/p$ where p is the period of run R)
- Sum of exponents of runs is $\mathcal{O}(n) \Rightarrow \mathcal{O}(n)$ chunks in total
  [Kolpakov and Kucherov, FOCS 99]
- Total length of run seed masks is $\mathcal{O}(n \log n)$
  $\Rightarrow$ they can be packed in $\mathcal{O}(n)$ machine words

Our Idea 3. Bit Parallelism

Power of word-RAM with $w = \Omega(\log n)$:

- Order runs by growing periods
- For the shortest cover of $rot_i(S)$

  only the first encounter of a 1 bit in a run seed mask at position i counts

  $\Rightarrow \mathcal{O}(n)$ assignments
- Use or implement `ffs` (find first bit set) to find such bits

sh. cov. computed: 0 1 1 0 0 1 1 1 0 0 1

$0 \quad 1 \quad \ldots \quad n-1 \quad n \quad \ldots \quad 2n-1 \; 2n \quad \ldots \quad 3n-1$

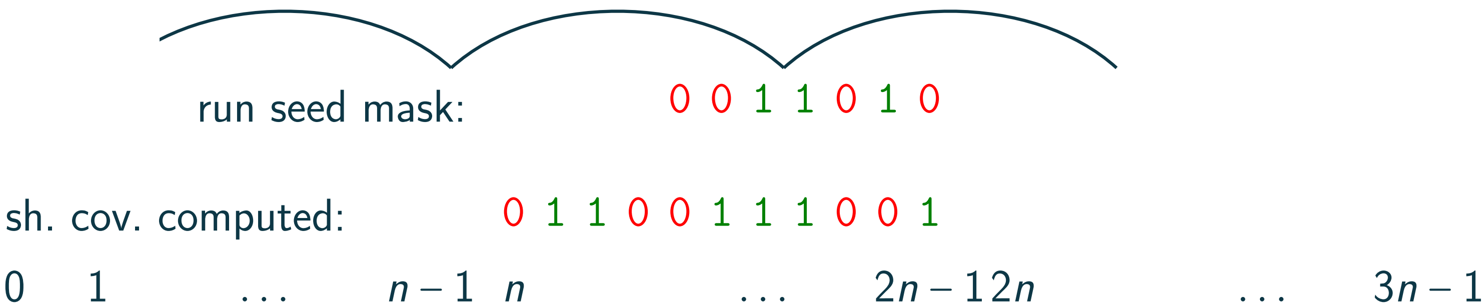
Our Idea 3. Bit Parallelism

Power of word-RAM with $w = \Omega(\log n)$:

- Order runs by growing periods
- For the shortest cover of $rot_i(S)$

  only the first encounter of a 1 bit in a run seed mask at position i counts

  $\Rightarrow \mathcal{O}(n)$ assignments
- Use or implement `ffs` (find first bit set) to find such bits



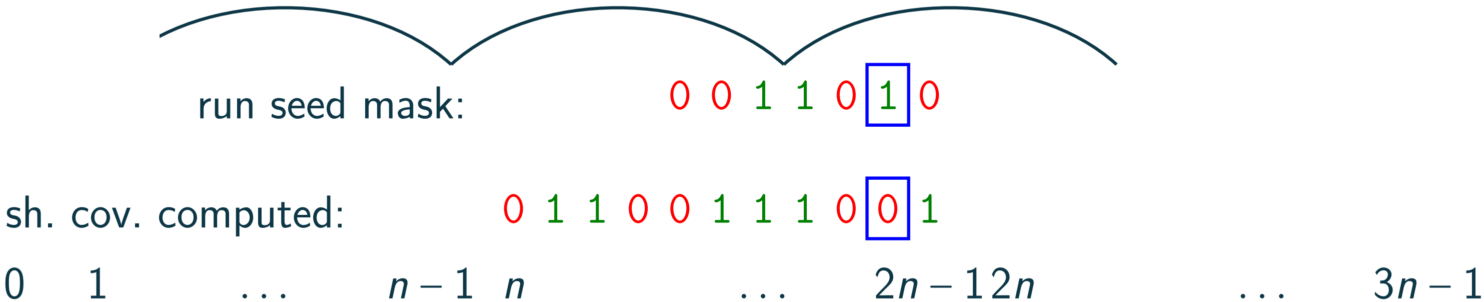
Our Idea 3. Bit Parallelism

Power of word-RAM with $w = \Omega(\log n)$:

- Order runs by growing periods
- For the shortest cover of $rot_i(S)$

  only the first encounter of a 1 bit in a run seed mask at position i counts

  $\Rightarrow \mathcal{O}(n)$ assignments
- Use or implement `ffs` (find first bit set) to find such bits



Conclusions

Theorem (This paper)

Shortest covers of all rotations of S over an integer alphabet can be computed in $\mathcal{O}(n)$ time in the word-RAM.

Conclusions

Theorem (This paper)

Shortest covers of all rotations of S over an integer alphabet can be computed in $\mathcal{O}(n)$ time in the word-RAM.

Conclusions

Theorem (This paper)

Shortest covers of all rotations of S over an integer alphabet can be computed in $\mathcal{O}(n)$ time in the word-RAM.

Question: Is there a simple linear time algorithm for any alphabet?

Conclusions

Theorem (This paper)

Shortest covers of all rotations of S over an integer alphabet can be computed in $\mathcal{O}(n)$ time in the word-RAM.

Question: Is there a simple linear time algorithm for any alphabet?

Cf. Breslauer's algorithm:

$B :=$ border array of S

for $i := 1$ **to** n **do**

if $(B[i] > 0)$ **and** $(R[C[B[i]]] \geq i - C[B[i]])$ **then**

$C[i] := C[B[i]]; \quad R[C[i]] := i$

else

$C[i] := R[i] := i$