Efficient Algorithms for Shortest Partial Seeds in Words

Tomasz Kociumaka[1], Solon P. Pissis[2],
Jakub Radoszewski[1], Wojciech Rytter[1],
Tomasz Waleń[1]

[1]University of Warsaw
[2]King's College London

CPM 2014

Moscow, June 16, 2014

Periodicity and quasiperiodicity

Periodicity: occurrences are aligned

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

Periodicity and quasiperiodicity

Periodicity: occurrences are aligned

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

Periodicity and quasiperiodicity

Periodicity: occurrences are aligned

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

Quasiperiodicity: occurrences may overlap

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

Periodicity and quasiperiodicity

Periodicity: occurrences are aligned

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

Quasiperiodicity: occurrences may overlap

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

Covers and seeds

Definition (Apostolico, Farach, Iliopoulos; 1991)

A factor u is a *cover* of w if each position (letter) in w lies within an occurrence of u in w.

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

Covers and seeds

Definition (Apostolico, Farach, Iliopoulos; 1991)

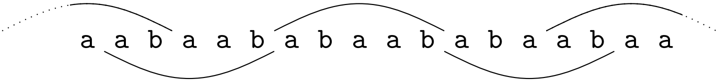
A factor u is a *cover* of w if each position (letter) in w lies within an occurrence of u in w.

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

Covers and seeds

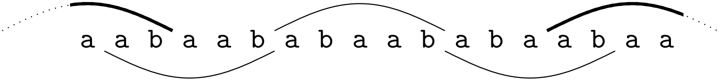
Definition (Iliopoulos, Moore, Park; 1993)

A factor u is a *seed* of w if u is a cover of a superstring of w.



Definition (Iliopoulos, Moore, Park; 1993)

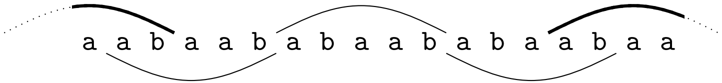
A factor u is a *seed* of w if u is a cover of a superstring of w.



Covers and seeds

Definition (Iliopoulos, Moore, Park; 1993)

A factor u is a *seed* of w if u is a cover of a superstring of w.



Observation

A factor u is a seed of w iff each position (letter) in w lies within a possibly overhanging occurrence of u in w.

Partial covers and partial seeds

Definition (KPRRW; CPM'13)

The *cover index* $\mathcal{C}(u)$ of u in w is the number of positions of w lying within an occurrence of u in w.

$$\texttt{a b a a b a a a b a a a a b a a a b a b a a a b}$$

$$\mathcal{C}(\texttt{a}) = 17$$

Partial covers and partial seeds

Definition (KPRRW; CPM'13)

The *cover index* $\mathcal{C}(u)$ of u in w is the number of positions of w lying within an occurrence of u in w.

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

$$\mathcal{C}(\mathtt{a}) = 17$$
$$\mathcal{C}(\mathtt{abaa}) = 19$$

Partial covers and partial seeds

Definition (KPRRW; CPM'13)

The *cover index* $\mathcal{C}(u)$ of u in w is the number of positions of w lying within an occurrence of u in w.

$$\texttt{a b a} \overbrace{\texttt{a b a a a b a}} \texttt{a a} \overbrace{\texttt{a b a a a b a}} \texttt{b a a a b}$$

$$\mathcal{C}(\texttt{a}) = 17$$
$$\mathcal{C}(\texttt{abaa}) = 19$$
$$\mathcal{C}(\texttt{abaaaba}) = 14$$

Partial covers and partial seeds

Definition (KPRRW; CPM'13)

The *cover index* $\mathcal{C}(u)$ of u in w is the number of positions of w lying within an occurrence of u in w.

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

$$\mathcal{C}(\mathtt{a}) = 17$$
$$\mathcal{C}(\mathtt{abaa}) = 19$$
$$\mathcal{C}(\mathtt{abaaaba}) = 14$$

Definition

For a positive integer α an α-partial cover of w is a factor of w with cover index at least α.

Partial covers and partial seeds

Definition

The *seed index* $\mathcal{S}(u)$ of u in w is the number of positions of w lying within a possibly overhanging occurrence of u in w.

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

$$\mathcal{C}(\mathtt{a}) = 17 \qquad \mathcal{S}(\mathtt{a}) = 17$$

$$\mathcal{C}(\mathtt{abaa}) = 19$$

$$\mathcal{C}(\mathtt{abaaaba}) = 14$$

Partial covers and partial seeds

Definition

The *seed index* $\mathcal{S}(u)$ of u in w is the number of positions of w lying within a possibly overhanging occurrence of u in w.

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

$$\mathcal{C}(\mathtt{a}) = 17$$
$$\mathcal{C}(\mathtt{abaa}) = 19$$
$$\mathcal{C}(\mathtt{abaaaba}) = 14$$

$$\mathcal{S}(\mathtt{a}) = 17$$
$$\mathcal{S}(\mathtt{abaa}) = 21$$

Partial covers and partial seeds

Definition

The *seed index* $\mathcal{S}(u)$ of u in w is the number of positions of w lying within a possibly overhanging occurrence of u in w.

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

$$\mathcal{C}(\text{a}) = 17 \qquad \mathcal{S}(\text{a}) = 17$$

$$\mathcal{C}(\text{abaa}) = 19 \qquad \mathcal{S}(\text{abaa}) = 21$$

$$\mathcal{C}(\text{abaaaba}) = 14 \qquad \mathcal{S}(\text{abaaaba}) = 22$$

Partial covers and partial seeds

Definition

The *seed index* $\mathcal{S}(u)$ of u in w is the number of positions of w lying within a possibly overhanging occurrence of u in w.

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

$$\mathcal{C}(\mathtt{a}) = 17 \qquad \mathcal{S}(\mathtt{a}) = 17$$
$$\mathcal{C}(\mathtt{abaa}) = 19 \qquad \mathcal{S}(\mathtt{abaa}) = 21$$
$$\mathcal{C}(\mathtt{abaaaba}) = 14 \qquad \mathcal{S}(\mathtt{abaaaba}) = 22$$

Definition

For a positive integer α an α-partial seed of w is a factor of w with seed index at least α.

Other variants of covers and seeds

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

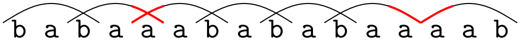
- *k-covers* and *k-seeds* (Iliopoulos, Smyth; 1998) – each position lies within a (possibly overhanging) occurrence of at least one of the few factors of length k, together forming a k-cover (k-seed).

Other variants of covers and seeds

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

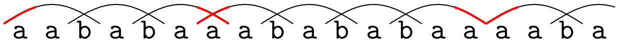
- *k-covers* and *k-seeds* (Iliopoulos, Smyth; 1998) – each position lies within a (possibly overhanging) occurrence of at least one of the few factors of length k, together forming a k-cover (k-seed).

Other variants of covers and seeds



- *k-covers* and *k-seeds* (Iliopoulos, Smyth; 1998) – each position lies within a (possibly overhanging) occurrence of at least one of the few factors of length k, together forming a k-cover (k-seed).
- *approximate covers* (Sim, Park, Kim, Lee; 2002) and *approximate seeds* (Christodoulakis et al.; 2005) – each position is lies within a (possibly overhanging) occurrence of a factor *similar* to the approximate cover (or seed).

Other variants of covers and seeds



- *k-covers* and *k-seeds* (Iliopoulos, Smyth; 1998) – each position lies within a (possibly overhanging) occurrence of at least one of the few factors of length k, together forming a k-cover (k-seed).
- *approximate covers* (Sim, Park, Kim, Lee; 2002) and *approximate seeds* (Christodoulakis et al.; 2005) – each position is lies within a (possibly overhanging) occurrence of a factor *similar* to the approximate cover (or seed).

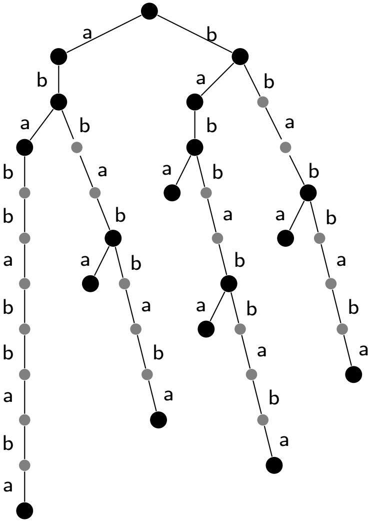
Other variants of covers and seeds

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

- *k-covers* and *k-seeds* (Iliopoulos, Smyth; 1998) – each position lies within a (possibly overhanging) occurrence of at least one of the few factors of length k, together forming a k-cover (k-seed).
- *approximate covers* (Sim, Park, Kim, Lee; 2002) and *approximate seeds* (Christodoulakis et al.; 2005) – each position is lies within a (possibly overhanging) occurrence of a factor *similar* to the approximate cover (or seed).

Main drawback: $\Omega(n^2)$ algorithms.

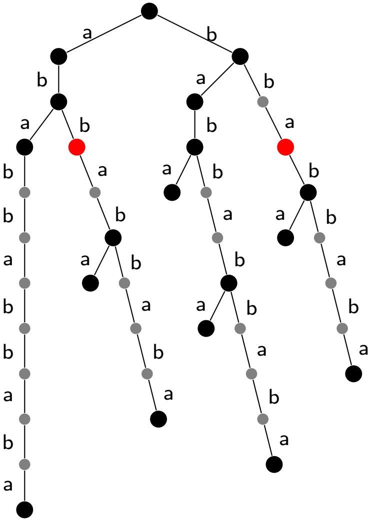
Cover Suffix Tree



CST(ababbabbaba)

The cover suffix tree of w (denoted $CST(w)$) is a suffix tree

Cover Suffix Tree

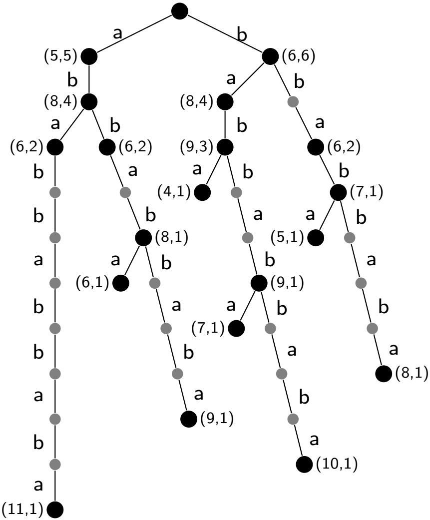


$CST(\texttt{ababbabbaba})$

The cover suffix tree of w (denoted $CST(w)$) is a suffix tree

- augmented with $\mathcal{O}(n)$ extra nodes,

Cover Suffix Tree

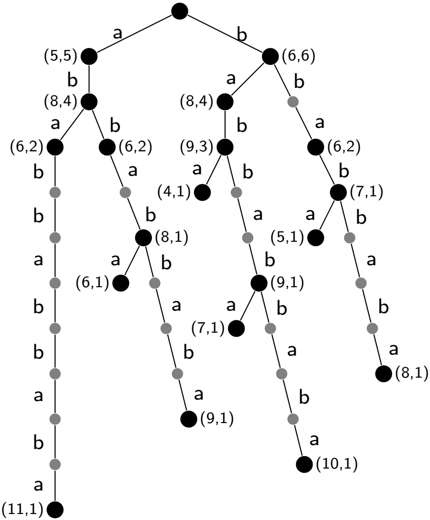


CST(ababbabbaba)

The cover suffix tree of w (denoted $CST(w)$) is a suffix tree

- augmented with $\mathcal{O}(n)$ extra nodes,
- with each node *annotated* with a pair of integers $(\mathcal{C}(v), \Delta(v))$.

Cover Suffix Tree



$CST(\texttt{ababbabbaba})$

The cover suffix tree of w (denoted $CST(w)$) is a suffix tree

- augmented with $\mathcal{O}(n)$ extra nodes,
- with each node *annotated* with a pair of integers $(\mathcal{C}(v), \Delta(v))$.

Theorem (KPRRW; CPM'13)

The tree $CST(w)$ can be built in $\mathcal{O}(n \log n)$ time for any word w of length n.

Our results

Problem (PARTIAL SEEDS)

Given a word w of length n and a positive integer $\alpha \leq n$ find all shortest factors u of w such that $\mathcal{S}(u) \geq \alpha$.

Problem (LIMITED LENGTH PARTIAL SEEDS)

Given a word w of length n and an interval $[\ell, r]$ find a factor u of w maximizing $\mathcal{S}(u)$ among factors for which $|u| \in [\ell, r]$.

Our results

Problem (Partial Seeds)

Given a word w of length n and a positive integer $\alpha \le n$ find all shortest factors u of w such that $\mathcal{S}(u) \ge \alpha$.

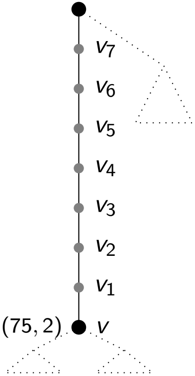
Problem (Limited Length Partial Seeds)

Given a word w of length n and an interval $[\ell, r]$ find a factor u of w maximizing $\mathcal{S}(u)$ among factors for which $|u| \in [\ell, r]$.

Theorem

Given $CST(w)$ both Partial Seeds *and* Limited Length Partial Seeds *can be solved in linear time.*

Determining the cover index

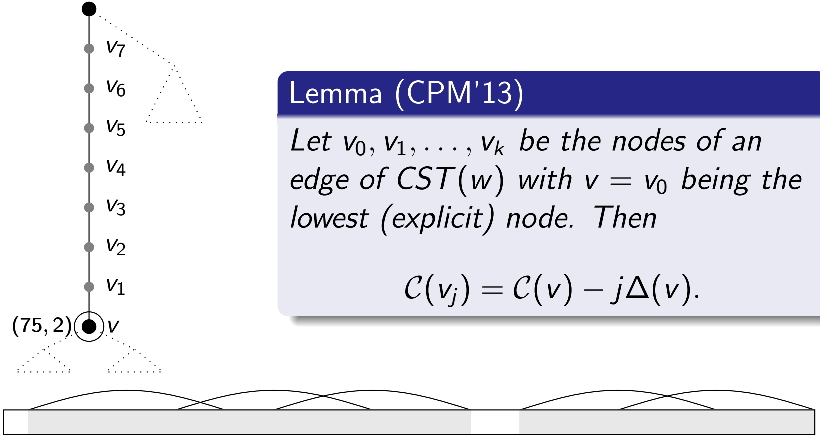


Lemma (CPM'13)

Let $v_0, v_1, \ldots, v_k$ be the nodes of an edge of $CST(w)$ with $v = v_0$ being the lowest (explicit) node. Then

$$\mathcal{C}(v_j) = \mathcal{C}(v) - j\Delta(v).$$

Determining the cover index

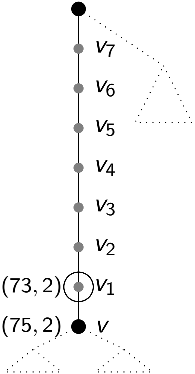


Lemma (CPM'13)

Let $v_0, v_1, \ldots, v_k$ be the nodes of an edge of $CST(w)$ with $v = v_0$ being the lowest (explicit) node. Then

$$\mathcal{C}(v_j) = \mathcal{C}(v) - j\Delta(v).$$

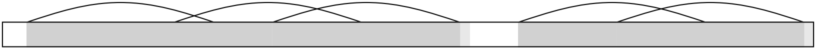
Determining the cover index



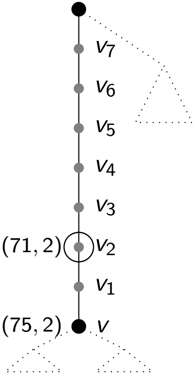
Lemma (CPM'13)

Let $v_0, v_1, \ldots, v_k$ be the nodes of an edge of $CST(w)$ with $v = v_0$ being the lowest (explicit) node. Then

$$\mathcal{C}(v_j) = \mathcal{C}(v) - j\Delta(v).$$



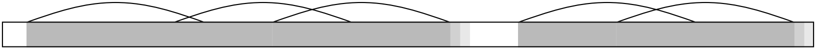
Determining the cover index



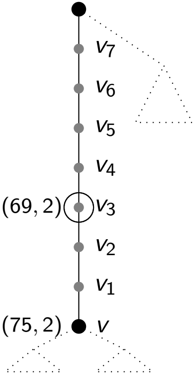
Lemma (CPM'13)

Let $v_0, v_1, \ldots, v_k$ be the nodes of an edge of $CST(w)$ with $v = v_0$ being the lowest (explicit) node. Then

$$\mathcal{C}(v_j) = \mathcal{C}(v) - j\Delta(v).$$



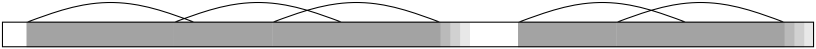
Determining the cover index



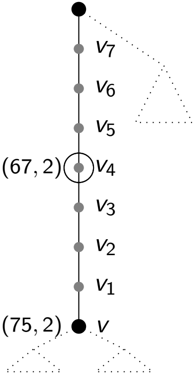
Lemma (CPM'13)

Let $v_0, v_1, \ldots, v_k$ be the nodes of an edge of $CST(w)$ with $v = v_0$ being the lowest (explicit) node. Then

$$\mathcal{C}(v_j) = \mathcal{C}(v) - j\Delta(v).$$



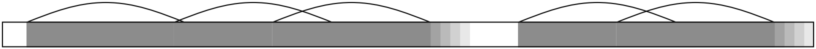
Determining the cover index



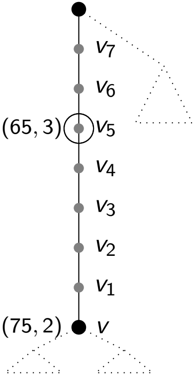
Lemma (CPM'13)

Let $v_0, v_1, \ldots, v_k$ be the nodes of an edge of $CST(w)$ with $v = v_0$ being the lowest (explicit) node. Then

$$\mathcal{C}(v_j) = \mathcal{C}(v) - j\Delta(v).$$



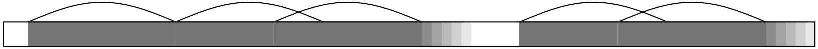
Determining the cover index



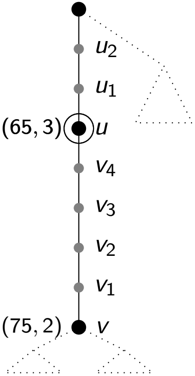
Lemma (CPM'13)

Let $v_0, v_1, \ldots, v_k$ be the nodes of an edge of $CST(w)$ with $v = v_0$ being the lowest (explicit) node. Then

$$\mathcal{C}(v_j) = \mathcal{C}(v) - j\Delta(v).$$



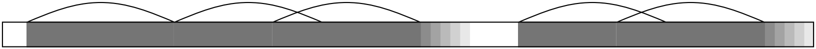
Determining the cover index



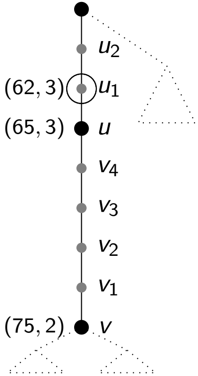
Lemma (CPM'13)

Let $v_0, v_1, \ldots, v_k$ be the nodes of an edge of $CST(w)$ with $v = v_0$ being the lowest (explicit) node. Then

$$\mathcal{C}(v_j) = \mathcal{C}(v) - j\Delta(v).$$



Determining the cover index

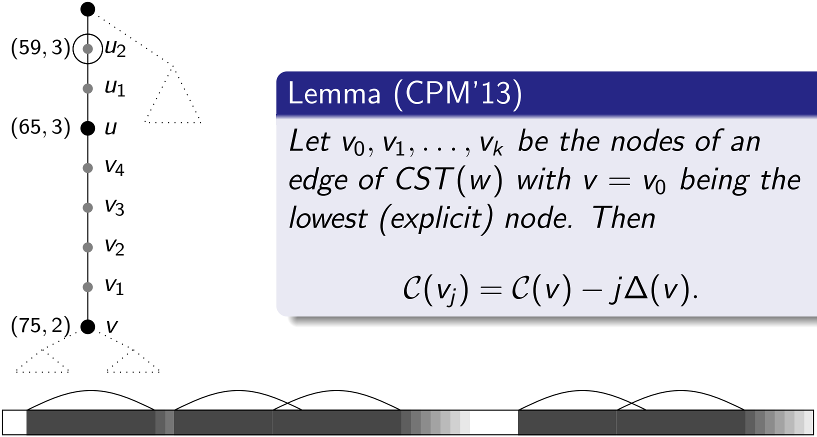


Lemma (CPM'13)

Let $v_0, v_1, \ldots, v_k$ be the nodes of an edge of $CST(w)$ with $v = v_0$ being the lowest (explicit) node. Then

$$\mathcal{C}(v_j) = \mathcal{C}(v) - j\Delta(v).$$

Determining the cover index

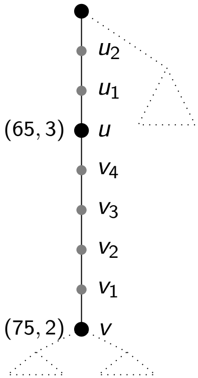


Lemma (CPM'13)

Let $v_0, v_1, \ldots, v_k$ be the nodes of an edge of $CST(w)$ with $v = v_0$ being the lowest (explicit) node. Then

$$\mathcal{C}(v_j) = \mathcal{C}(v) - j\Delta(v).$$

Determining the cover index



Lemma (CPM'13)

Let $v_0, v_1, \ldots, v_k$ be the nodes of an edge of $CST(w)$ with $v = v_0$ being the lowest (explicit) node. Then

$$\mathcal{C}(v_j) = \mathcal{C}(v) - j\Delta(v).$$

Corollary

Given a locus of v in $CST(w)$, the cover index $\mathcal{C}(v)$ can be computed in $\mathcal{O}(1)$ time.

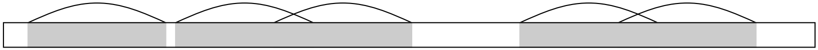
Seed index

$$\mathcal{S}(v) \quad = \qquad\qquad\qquad + \qquad\qquad + $$

Seed index

$$\mathcal{S}(v) \quad = \qquad\qquad\qquad + \quad \mathcal{C}(v) \quad +$$

Full occs

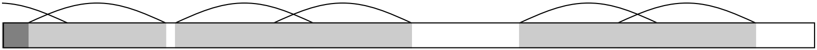


Seed index

$$\mathcal{S}(v) \quad = \quad \mathit{Left}\mathcal{S}(v) \quad + \quad \mathcal{C}(v) \quad +$$

Left-overhanging occs only

Full occs



Seed index

$$\mathcal{S}(v) \quad = \quad \mathit{Left}\mathcal{S}(v) \quad + \quad \mathcal{C}(v) \quad + \quad \mathit{Right}\mathcal{S}(v)$$

Left-overhanging occs only

Full occs

Right-overhanging occs only

Seed index

$$\mathcal{S}(v) \quad = \quad \mathit{Left}\mathcal{S}(v) \quad + \quad \mathcal{C}(v) \quad + \quad \mathit{Right}\mathcal{S}(v)$$

Left-overhanging occs only

Full occs

Right-overhanging occs only

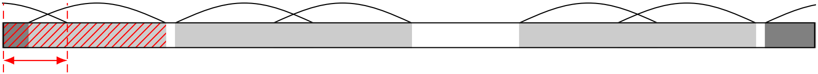
$$\mathit{Left}\mathcal{S}(v) = \min(B[\mathit{first}(v) + |v| - 1], \mathit{first}(v) - 1)$$

$\mathit{first}(v)$ start position of the first occurrence of v,

$B[i]$ largest border of $w[1..i]$.

Seed index

$$\mathcal{S}(v) \quad = \quad \underbrace{Left\mathcal{S}(v)}_{\text{Left-overhanging occs only}} \quad + \quad \underbrace{\mathcal{C}(v)}_{\text{Full occs}} \quad + \quad \underbrace{Right\mathcal{S}(v)}_{\text{Right-overhanging occs only}}$$



$$Left\mathcal{S}(v) = \min(B[first(v) + |v| - 1], first(v) - 1)$$

$first(v)$ start position of the first occurrence of v,

$B[i]$ largest border of $w[1..i]$.

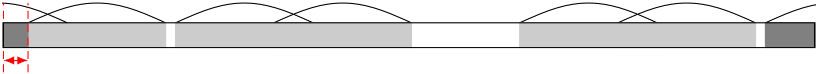
Seed index

$$\mathcal{S}(v) \quad = \quad Left\mathcal{S}(v) \quad + \quad \mathcal{C}(v) \quad + \quad Right\mathcal{S}(v)$$

Left-overhanging occs only

Full occs

Right-overhanging occs only



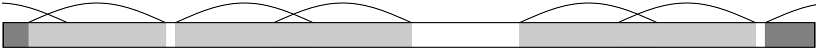
$$Left\mathcal{S}(v) = \min(B[first(v) + |v| - 1], first(v) - 1)$$

$first(v)$ start position of the first occurrence of v,

$B[i]$ largest border of $w[1..i]$.

Seed index

$$\mathcal{S}(v) \quad = \quad \underbrace{\mathit{Left}\mathcal{S}(v)}_{\text{Left-overhanging occs only}} \quad + \quad \underbrace{\mathcal{C}(v)}_{\text{Full occs}} \quad + \quad \underbrace{\mathit{Right}\mathcal{S}(v)}_{\text{Right-overhanging occs only}}$$



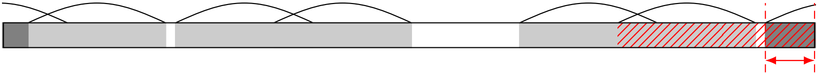
$$\mathit{Left}\mathcal{S}(v) = \min(B[\mathit{first}(v) + |v| - 1], \mathit{first}(v) - 1)$$
$$\mathit{Right}\mathcal{S}(v) = \min(B^R[\mathit{last}(v)], n - |v| + 1 - \mathit{last}(v))$$

$\mathit{last}(v)$ start position of the last occurrence of v,

$B^R[i]$ largest border of $w[i..n]$.

Seed index

$$\mathcal{S}(v) = \underbrace{\mathit{Left}\mathcal{S}(v)}_{\text{Left-overhanging occs only}} + \underbrace{\mathcal{C}(v)}_{\text{Full occs}} + \underbrace{\mathit{Right}\mathcal{S}(v)}_{\text{Right-overhanging occs only}}$$



$$\mathit{Left}\mathcal{S}(v) = \min(B[\mathit{first}(v) + |v| - 1], \mathit{first}(v) - 1)$$

$$\mathit{Right}\mathcal{S}(v) = \min(B^R[\mathit{last}(v)], n - |v| + 1 - \mathit{last}(v))$$

$\mathit{last}(v)$ start position of the last occurrence of v,

$B^R[i]$ largest border of $w[i..n]$.

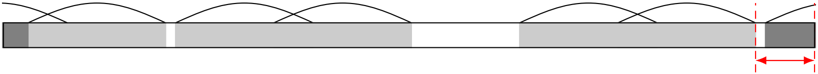
Seed index

$$\mathcal{S}(v) \quad = \quad \mathit{Left}\mathcal{S}(v) \quad + \quad \mathcal{C}(v) \quad + \quad \mathit{Right}\mathcal{S}(v)$$

Left-overhanging occs only

Full occs

Right-overhanging occs only



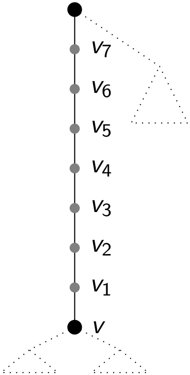
$$\mathit{Left}\mathcal{S}(v) = \min(B[\mathit{first}(v) + |v| - 1], \mathit{first}(v) - 1)$$

$$\mathit{Right}\mathcal{S}(v) = \min(B^R[\mathit{last}(v)], n - |v| + 1 - \mathit{last}(v))$$

$\mathit{last}(v)$ start position of the last occurrence of v,

$B^R[i]$ largest border of $w[i..n]$.

Seed index on an edge of $CST(w)$



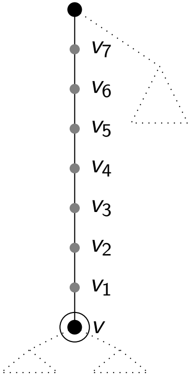
$$\mathcal{S}(v_j) = \mathit{Left}\mathcal{S}(v_j) + \mathcal{C}(v_j) + \mathit{Right}\mathcal{S}(v_j)$$

$\mathcal{C}(v_j)$ a linear function,

$\mathit{Right}\mathcal{S}(v_j)$ becomes a linear upon creation of ≤ 1 extra node per edge,

$\mathit{Left}\mathcal{S}(v_j)$ less regular: $\min(B[r_v - j], c_v)$.

Seed index on an edge of $CST(w)$

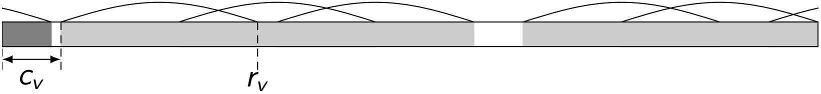


$$\mathcal{S}(v_j) = Left\mathcal{S}(v_j) + \mathcal{C}(v_j) + Right\mathcal{S}(v_j)$$

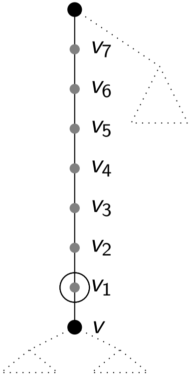
$\mathcal{C}(v_j)$ a linear function,

$Right\mathcal{S}(v_j)$ becomes a linear upon creation of ≤ 1 extra node per edge,

$Left\mathcal{S}(v_j)$ less regular: $\min(B[r_v - j], c_v)$.



Seed index on an edge of $CST(w)$

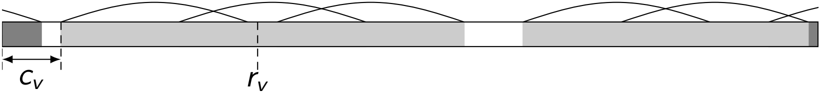


$$\mathcal{S}(v_j) = Left\mathcal{S}(v_j) + \mathcal{C}(v_j) + Right\mathcal{S}(v_j)$$

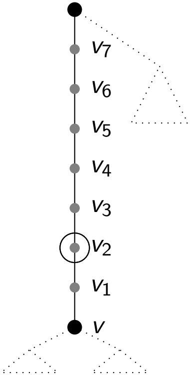
$\mathcal{C}(v_j)$ a linear function,

$Right\mathcal{S}(v_j)$ becomes a linear upon creation of ≤ 1 extra node per edge,

$Left\mathcal{S}(v_j)$ less regular: $\min(B[r_v - j], c_v)$.



Seed index on an edge of $CST(w)$

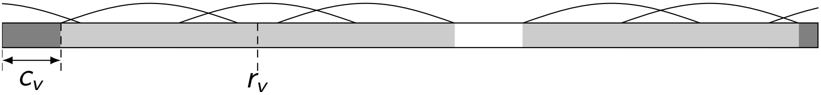


$$\mathcal{S}(v_j) = Left\mathcal{S}(v_j) + \mathcal{C}(v_j) + Right\mathcal{S}(v_j)$$

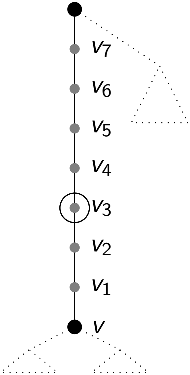
$\mathcal{C}(v_j)$ a linear function,

$Right\mathcal{S}(v_j)$ becomes a linear upon creation of ≤ 1 extra node per edge,

$Left\mathcal{S}(v_j)$ less regular: $\min(B[r_v - j], c_v)$.



Seed index on an edge of $CST(w)$

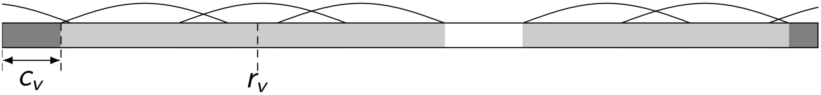


$$\mathcal{S}(v_j) = Left\mathcal{S}(v_j) + \mathcal{C}(v_j) + Right\mathcal{S}(v_j)$$

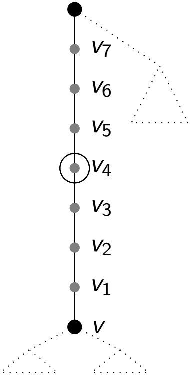
$\mathcal{C}(v_j)$ a linear function,

$Right\mathcal{S}(v_j)$ becomes a linear upon creation of ≤ 1 extra node per edge,

$Left\mathcal{S}(v_j)$ less regular: $\min(B[r_v - j], c_v)$.



Seed index on an edge of $CST(w)$

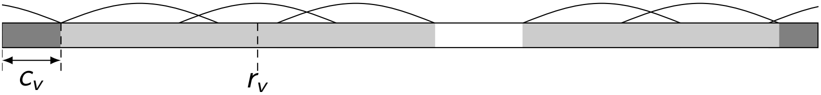


$$\mathcal{S}(v_j) = Left\mathcal{S}(v_j) + \mathcal{C}(v_j) + Right\mathcal{S}(v_j)$$

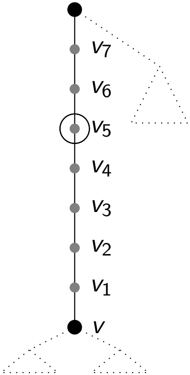
$\mathcal{C}(v_j)$ a linear function,

$Right\mathcal{S}(v_j)$ becomes a linear upon creation of ≤ 1 extra node per edge,

$Left\mathcal{S}(v_j)$ less regular: $\min(B[r_v - j], c_v)$.



Seed index on an edge of $CST(w)$

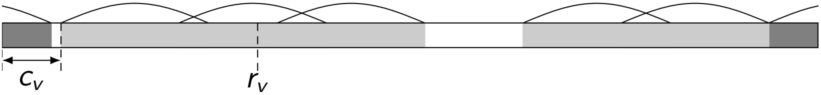


$$\mathcal{S}(v_j) = Left\mathcal{S}(v_j) + \mathcal{C}(v_j) + Right\mathcal{S}(v_j)$$

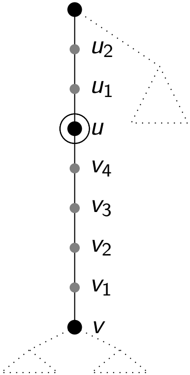
$\mathcal{C}(v_j)$ a linear function,

$Right\mathcal{S}(v_j)$ becomes a linear upon creation of ≤ 1 extra node per edge,

$Left\mathcal{S}(v_j)$ less regular: $\min(B[r_v - j], c_v)$.



Seed index on an edge of $CST(w)$

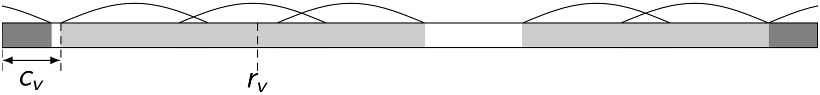


$$\mathcal{S}(v_j) = Left\mathcal{S}(v_j) + \mathcal{C}(v_j) + Right\mathcal{S}(v_j)$$

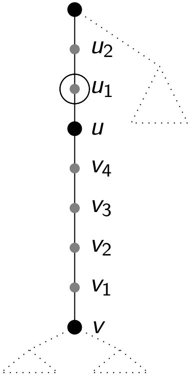
$\mathcal{C}(v_j)$ a linear function,

$Right\mathcal{S}(v_j)$ becomes a linear upon creation of ≤ 1 extra node per edge,

$Left\mathcal{S}(v_j)$ less regular: $\min(B[r_v - j], c_v)$.



Seed index on an edge of $CST(w)$

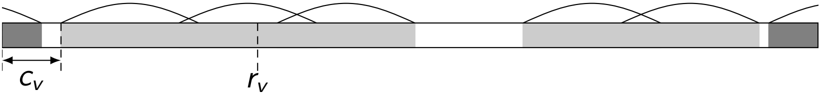


$$\mathcal{S}(v_j) = Left\mathcal{S}(v_j) + \mathcal{C}(v_j) + Right\mathcal{S}(v_j)$$

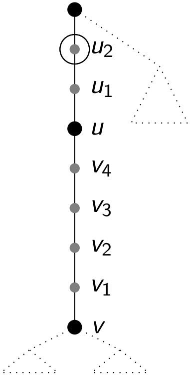
$\mathcal{C}(v_j)$ a linear function,

$Right\mathcal{S}(v_j)$ becomes a linear upon creation of ≤ 1 extra node per edge,

$Left\mathcal{S}(v_j)$ less regular: $\min(B[r_v - j], c_v)$.



Seed index on an edge of $CST(w)$

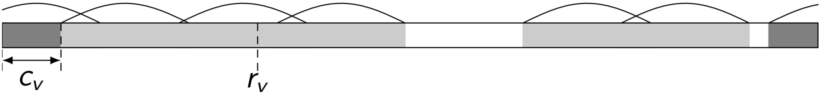


$$\mathcal{S}(v_j) = Left\mathcal{S}(v_j) + \mathcal{C}(v_j) + Right\mathcal{S}(v_j)$$

$\mathcal{C}(v_j)$ a linear function,

$Right\mathcal{S}(v_j)$ becomes a linear upon creation of ≤ 1 extra node per edge,

$Left\mathcal{S}(v_j)$ less regular: $\min(B[r_v - j], c_v)$.



Seed Suffix Tree

$CST(w)$ can be further augmented in $\mathcal{O}(n)$ time to $SST(w)$ (*Seed Suffix Tree*) such that

- for each node v there exists a function

$$\phi_v(x) = a_v x + b_v + \min(c_v, B[x])$$

and a range $R_v = (\ell_v, r_v]$ such that $\mathcal{S}(v_j) = \phi_v(r_v - j)$ for any v_j on the edge immediately above v,

Seed Suffix Tree

$CST(w)$ can be further augmented in $\mathcal{O}(n)$ time to $SST(w)$ (*Seed Suffix Tree*) such that

- for each node v there exists a function

  $$\phi_v(x) = a_v x + b_v + \min(c_v, B[x])$$

  and a range $R_v = (\ell_v, r_v]$ such that $\mathcal{S}(v_j) = \phi_v(r_v - j)$ for any v_j on the edge immediately above v,
- $0 \le a_v \le \mathrm{Occ}(v)$, where $\mathrm{Occ}(v)$ is the number of occurrences of v in w.

Seed Suffix Tree

$CST(w)$ can be further augmented in $\mathcal{O}(n)$ time to $SST(w)$ (*Seed Suffix Tree*) such that

- for each node v there exists a function

$$\phi_v(x) = a_v x + b_v + \min(c_v, B[x])$$

  and a range $R_v = (\ell_v, r_v]$ such that $\mathcal{S}(v_j) = \phi_v(r_v - j)$ for any v_j on the edge immediately above v,
- $0 \leq a_v \leq \mathrm{Occ}(v)$, where $\mathrm{Occ}(v)$ is the number of occurrences of v in w.

Observation

Given a locus of v in $SST(w)$ and the border table B, the seed index $\mathcal{S}(v)$ can be computed in $\mathcal{O}(1)$ time.

Abstract problems

Problem

Input: *pairs* (ϕ_i, R_i), *where* $\phi_i(x) = a_i x + b_i + \min(c_i, B[x])$ *is a function and* $R_i = (\ell_i, r_i] \subseteq [1, n]$ *is a non-empty range*
Output:

(a) $\operatorname{argmax}\{\phi_i(x) : x \in R_i\}$ *for each pair,*

(b) $\min\{x \in R_i : \phi_i(x) \geq \alpha\}$ *for each pair.*

Abstract problems

Problem

Input: *pairs* (ϕ_i, R_i), *where* $\phi_i(x) = a_i x + b_i + \min(c_i, B[x])$ *is a function and* $R_i = (\ell_i, r_i] \subseteq [1, n]$ *is a non-empty range*
Output:

(a) $\operatorname{argmax}\{\phi_i(x) : x \in R_i\}$ *for each pair,*

(b) $\min\{x \in R_i : \phi_i(x) \geq \alpha\}$ *for each pair.*

Lemma

Values (a) and (b) can be computed (offline) in linear time.

Abstract problems

Problem

Input: *pairs* (ϕ_i, R_i), *where* $\phi_i(x) = a_i x + b_i + \min(c_i, B[x])$ *is a function and* $R_i = (\ell_i, r_i] \subseteq [1, n]$ *is a non-empty range*
Output:

(a) $\operatorname{argmax}\{\phi_i(x) : x \in R_i\}$ *for each pair,*

(b) $\min\{x \in R_i : \phi_i(x) \geq \alpha\}$ *for each pair.*

Lemma

Values (a) and (b) can be computed (offline) in linear time.
Additional assumption required for (b): $\sum a_i = \mathcal{O}(n)$.

Abstract problems

Problem

Input: *pairs* (ϕ_i, R_i), *where* $\phi_i(x) = a_i x + b_i + \min(c_i, B[x])$ *is a function and* $R_i = (\ell_i, r_i] \subseteq [1, n]$ *is a non-empty range*
Output:

(a) $\operatorname{argmax}\{\phi_i(x) : x \in R_i\}$ *for each pair,*

(b) $\min\{x \in R_i : \phi_i(x) \geq \alpha\}$ *for each pair.*

Lemma

Values (a) and (b) can be computed (offline) in linear time.
Additional assumption required for (b): $\sum a_i = \mathcal{O}(n)$.

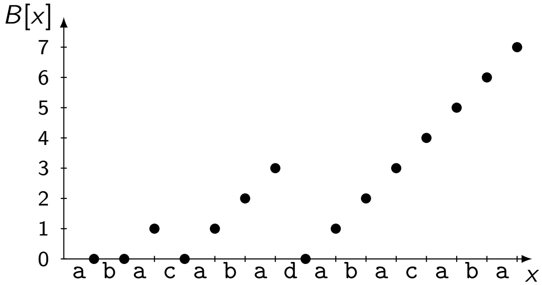
Workaround for $\sum a_i = \mathcal{O}(n)$:

- use (a) queries to restrict the set of edges queried for (b).

Toy problem

Problem

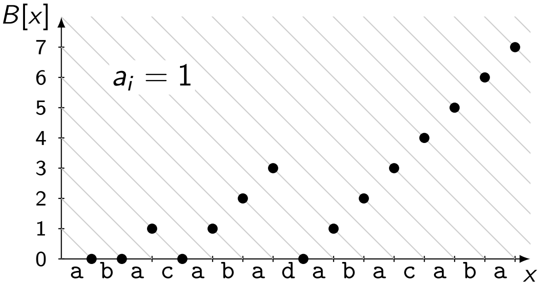
For the border array B to answer (off-line) the following queries: given a non-negative coefficient a_i and a range $R_i = (\ell_i, r_i]$ compute $x_i = \text{argmax}\{a_i x + B[x] : x \in R_i\}$.



Toy problem

Problem

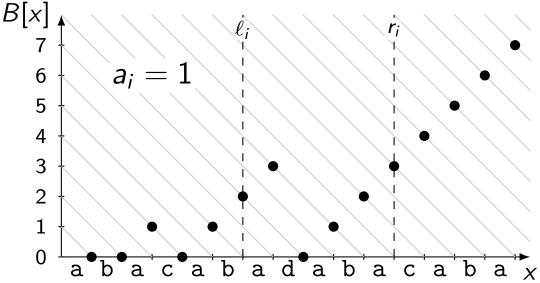
For the border array B to answer (off-line) the following queries: given a non-negative coefficient a_i and a range $R_i = (\ell_i, r_i]$ compute $x_i = \operatorname{argmax}\{a_i x + B[x] : x \in R_i\}$.



Toy problem

Problem

For the border array B to answer (off-line) the following queries: given a non-negative coefficient a_i and a range $R_i = (\ell_i, r_i]$ compute $x_i = \mathrm{argmax}\{a_i x + B[x] : x \in R_i\}$.



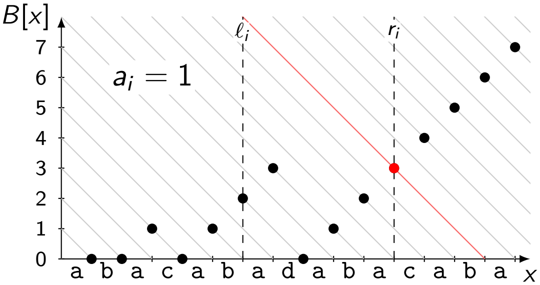
Observation

For each query we have $x_i = r_i$ or $B[x_i + 1] < B[x_i] - a_i$.

Toy problem

Problem

For the border array B to answer (off-line) the following queries: given a non-negative coefficient a_i and a range $R_i = (\ell_i, r_i]$ compute $x_i = \operatorname{argmax}\{a_i x + B[x] : x \in R_i\}$.



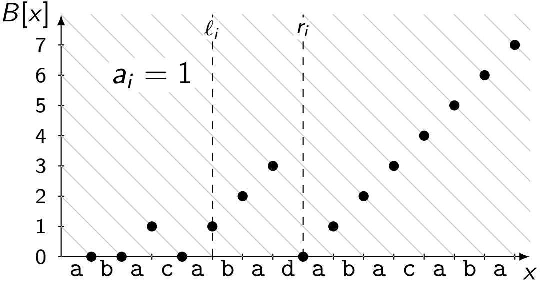
Observation

For each query we have $x_i = r_i$ or $B[x_i + 1] < B[x_i] - a_i$.

Toy problem

Problem

For the border array B to answer (off-line) the following queries: given a non-negative coefficient a_i and a range $R_i = (\ell_i, r_i]$ compute $x_i = \operatorname{argmax}\{a_i x + B[x] : x \in R_i\}$.



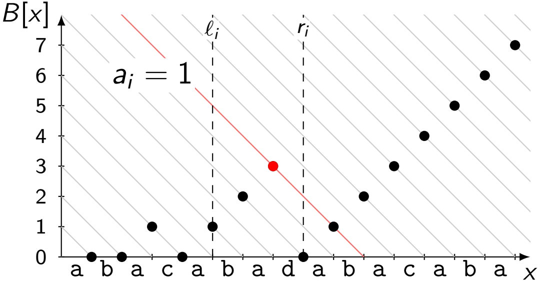
Observation

For each query we have $x_i = r_i$ or $B[x_i + 1] < B[x_i] - a_i$.

Toy problem

Problem

For the border array B to answer (off-line) the following queries: given a non-negative coefficient a_i and a range $R_i = (\ell_i, r_i]$ compute $x_i = \operatorname{argmax}\{a_i x + B[x] : x \in R_i\}$.



Observation

For each query we have $x_i = r_i$ or $B[x_i + 1] < B[x_i] - a_i$.

Toy problem: solution

Observation

Let $F_a = \{x : B[x+1] < B[x] - a\}$. *Then* $\sum_{a \geq 0} |F_a| = \mathcal{O}(n)$.

Toy problem: solution

Observation

Let $F_a = \{x : B[x+1] < B[x] - a\}$. *Then* $\sum_{a \geq 0} |F_a| = \mathcal{O}(n)$.

Proof.

$B[x+1] \leq B[x] + 1$, i.e. the total increase in B is at most n.
$\sum_{a \geq 0} |F_a|$ is bounded by the total decrease of in B. □

Toy problem: solution

Observation

Let $F_a = \{x : B[x+1] < B[x] - a\}$*. Then* $\sum_{a\geq 0} |F_a| = \mathcal{O}(n)$*.*

Proof.

$B[x+1] \leq B[x] + 1$, i.e. the total increase in B is at most n.
$\sum_{a\geq 0} |F_a|$ is bounded by the total decrease of in B. □

1. Apply (offline) predecessor queries to translate the range R_i into the range of positions in F_{a_i}.

Toy problem: solution

Observation

Let $F_a = \{x : B[x+1] < B[x] - a\}$*. Then* $\sum_{a \geq 0} |F_a| = \mathcal{O}(n)$*.*

Proof.

$B[x+1] \leq B[x] + 1$, i.e. the total increase in B is at most n.
$\sum_{a \geq 0} |F_a|$ is bounded by the total decrease of in B. □

1. Apply (offline) predecessor queries to translate the range R_i into the range of positions in F_{a_i}.
2. Use range maximum queries (RMQ) for $a_i x + B[x]$ and $x \in F_{a_i}$ to compute $\text{argmax}\{a_i x + B[x] : x \in R_i \cap F_{a_i}\}$.

Toy problem: solution

Observation

Let $F_a = \{x : B[x+1] < B[x] - a\}$. *Then* $\sum_{a \geq 0} |F_a| = \mathcal{O}(n)$.

Proof.

$B[x+1] \leq B[x] + 1$, i.e. the total increase in B is at most n.
$\sum_{a \geq 0} |F_a|$ is bounded by the total decrease of in B. □

1. Apply (offline) predecessor queries to translate the range R_i into the range of positions in F_{a_i}.
2. Use range maximum queries (RMQ) for $a_i x + B[x]$ and $x \in F_{a_i}$ to compute $\mathrm{argmax}\{a_i x + B[x] : x \in R_i \cap F_{a_i}\}$.
3. For each query check the possibility of $x_i = r_i$.

Conclusions and open problems

Two problems regarding partial seeds can be solved in $\mathcal{O}(n)$ time provided that $CST(w)$ is already computed:

Conclusions and open problems

Two problems regarding partial seeds can be solved in $\mathcal{O}(n)$ time provided that $CST(w)$ is already computed:

- find the shortest factor u with $\mathcal{S}(u)$ exceeding a given threshold α,

Conclusions and open problems

Two problems regarding partial seeds can be solved in $\mathcal{O}(n)$ time provided that $CST(w)$ is already computed:

- find the shortest factor u with $\mathcal{S}(u)$ exceeding a given threshold α,
- find the factor u maximizing $\mathcal{S}(u)$ and satisfying length restrictions,
    - other kinds of restrictions also possible.

Conclusions and open problems

Two problems regarding partial seeds can be solved in $\mathcal{O}(n)$ time provided that $CST(w)$ is already computed:

- find the shortest factor u with $\mathcal{S}(u)$ exceeding a given threshold α,
- find the factor u maximizing $\mathcal{S}(u)$ and satisfying length restrictions,
    - other kinds of restrictions also possible.

Open problems:

- improve the construction algorithm for $CST(w)$ (currently $\mathcal{O}(w \log n)$),

Conclusions and open problems

Two problems regarding partial seeds can be solved in $\mathcal{O}(n)$ time provided that $CST(w)$ is already computed:

- find the shortest factor u with $\mathcal{S}(u)$ exceeding a given threshold α,
- find the factor u maximizing $\mathcal{S}(u)$ and satisfying length restrictions,
  - other kinds of restrictions also possible.

Open problems:

- improve the construction algorithm for $CST(w)$ (currently $\mathcal{O}(w \log n)$),
- for each length find the factor u maximizing $\mathcal{S}(u)$
  - for partial covers $\mathcal{O}(n \log n)$ time.

Thank you for your attention!