Compressed String Dictionary Look-up with Edit Distance One

Djamal Belazzougui

LIAFA - Paris 7

Rossano Venturini

University of Pisa

Problem

- Design a (compressed) index for a set $D=\{S_1, S_2, ..., S_d\}$ of d strings of total length n drawn from an alphabet of size σ
- Given any pattern P, report all the occ strings in D having Edit Distance at most one from P
  - edit(P, S) = minimum # of edit operations (insertion, deletion, substitution) to transform P into S

Motivations

Motivations

- Information Retrieval

Motivations

- Information Retrieval
  - e.g., users may misspell terms in their queries
- Database
  - e.g., you may do not remember the correct spelling of a key
- Data Mining
  - e.g, clean a noised dataset

Known solutions

Known solutions

- G. Brodal and L. Gasieniec [CPM'96]
  - Space: O(σn log n) bits
  - Time: O(|P| + occ)

Known solutions

- G. Brodal and L. Gasieniec [CPM'96]
  - Space: O(σn log n) bits
  - Time: O(|P| + occ)

Reduced to membership queries
in a super set containing
all the strings that have
edit distance one from any string in D.

For each S∈D, we add O(σ|S|)

Known solutions

- G. Brodal and L. Gasieniec [CPM'96]
  - Space: O(σn log n) bits
  - Time: O(|P| + occ)
- G. Brodal and L. Gasieniec [CPM'96]
  - Space: O(n log n) bits
  - Time: O(|P| + log n + occ)

Known solutions

- G. Brodal and L. Gasieniec [CPM'96]
  - Space: O(σn log n) bits
  - Time: O(|P| + occ)
- G. Brodal and L. Gasieniec [CPM'96]
  - Space: O(n log n) bits
  - Time: O(|P| + log n + occ)

For each S∈D, consider its factorizations of the form

S[1,i-1] S[i] S[i+1,s]

Known solutions

- G. Brodal and L. Gasieniec [CPM'96]
  - Space: O(σn log n) bits
  - Time: O(|P| + occ)
- G. Brodal and L. Gasieniec [CPM'96]
  - Space: O(n log n) bits
  - Time: O(|P| + log n + occ)

For each S∈D, consider its factorizations of the form

S[1,i-1] S[i] S[i+1,s]

If P[1,i-1] == S[1,i-1]
and
P[i,p] == S[i+1,s], and s=p+1
then,
S has edit distance one from P

Known solutions

- G. Brodal and L. Gasieniec [CPM'96]
  - Space: O(σn log n) bits
  - Time: O(|P| + occ)
- G. Brodal and L. Gasieniec [CPM'96]
  - Space: O(n log n) bits
  - Time: O(|P| + log n + occ)

For each S∈D, consider its factorizations of the form

S[1,i-1] S[i] S[i+1,s]

If P[1,i-1] == S[1,i-1]
and
P[i,p] == S[i+1,s], and s=p+1
then,
S has edit distance one from P

Use Balanced Search Trees to index these factorizations

Known solutions

- G. Brodal and L. Gasieniec [CPM'96]
  - Space: O(σn log n) bits
  - Time: O(|P| + occ)
- G. Brodal and L. Gasieniec [CPM'96]
  - Space: O(n log n) bits
  - Time: O(|P| + log n + occ)
- D. Belazzougui [CPM'09]
  - Space: O(n log σ)
  - Time: O(|P| + occ)

Known solutions

- G. Brodal and L. Gasieniec [CPM'96]
  - Space: O(σn log n) bits
  - Time: O(|P| + occ)
- G. Brodal and L. Gasieniec [CPM'96]
  - Space: O(n log n) bits
  - Time: O(|P| + log n + occ)
- D. Belazzougui [CPM'09]
  - Space: O(n log σ)
  - Time: O(|P| + occ)

Similar approach

Use of Minimal Perfect
Hash Functions to index factorizations
and to save space

Our results

Our results

Solution based on Patricia tries and Perfect hashing

Time $O(|P| + occ)$

Space $2nH_k + n \cdot o(\log \sigma) + 2d \log d$

Our results

Solution based on Patricia tries and Perfect hashing

Time $O(|P| + occ)$

Space $2nH_k + n \cdot o(\log \sigma) + 2d \log d$

Solution based on Compressed Permuterm index

Time $O(|P| \log \log \sigma + occ)$

Space $nH_k + n \cdot o(\log \sigma)$ for $\sigma = \omega(\log^c n)$

Our results

Solution based on Patricia tries and Perfect hashing

Time $O(|P| + occ)$

Space $2nH_k + n \cdot o(\log \sigma) + 2d \log d$

Top-k strings in $O(|P| + k \log k)$

Solution based on Compressed Permuterm index

Time $O(|P| \log \log \sigma + occ)$

Space $nH_k + n \cdot o(\log \sigma)$ for $\sigma = \omega(\log^c n)$

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc accb baca caac cbcc}$$

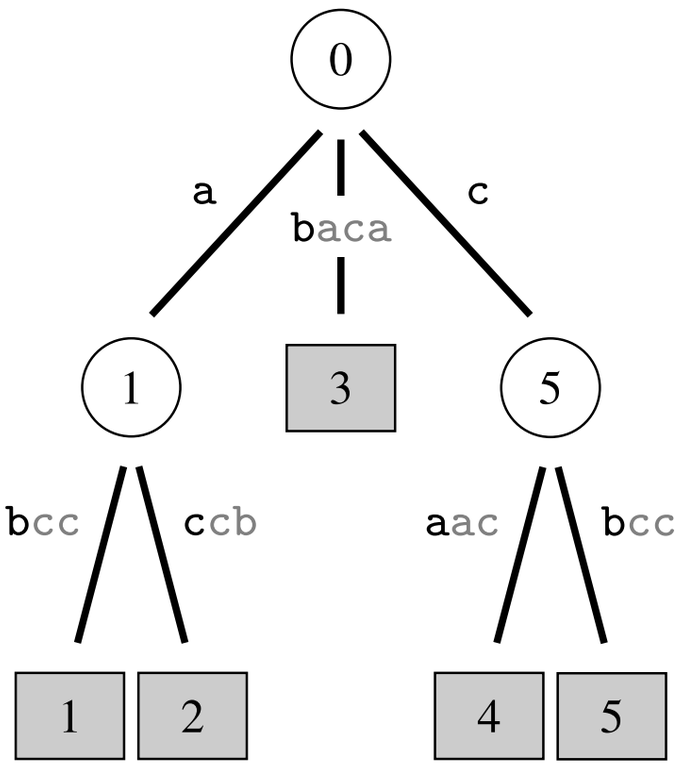
$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc accb baca caac cbcc}$$

access time: O(1)
space: H_k + n o(log σ)

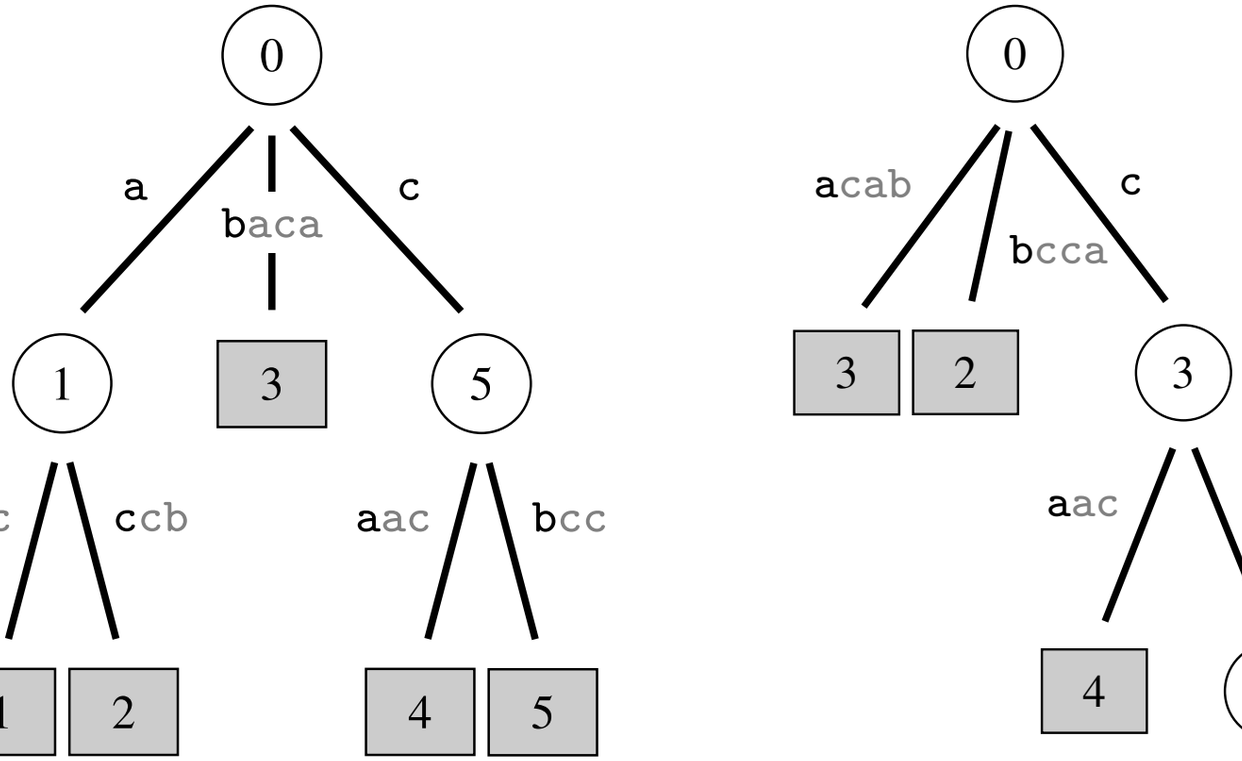
$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc accb baca caac cbcc}$$



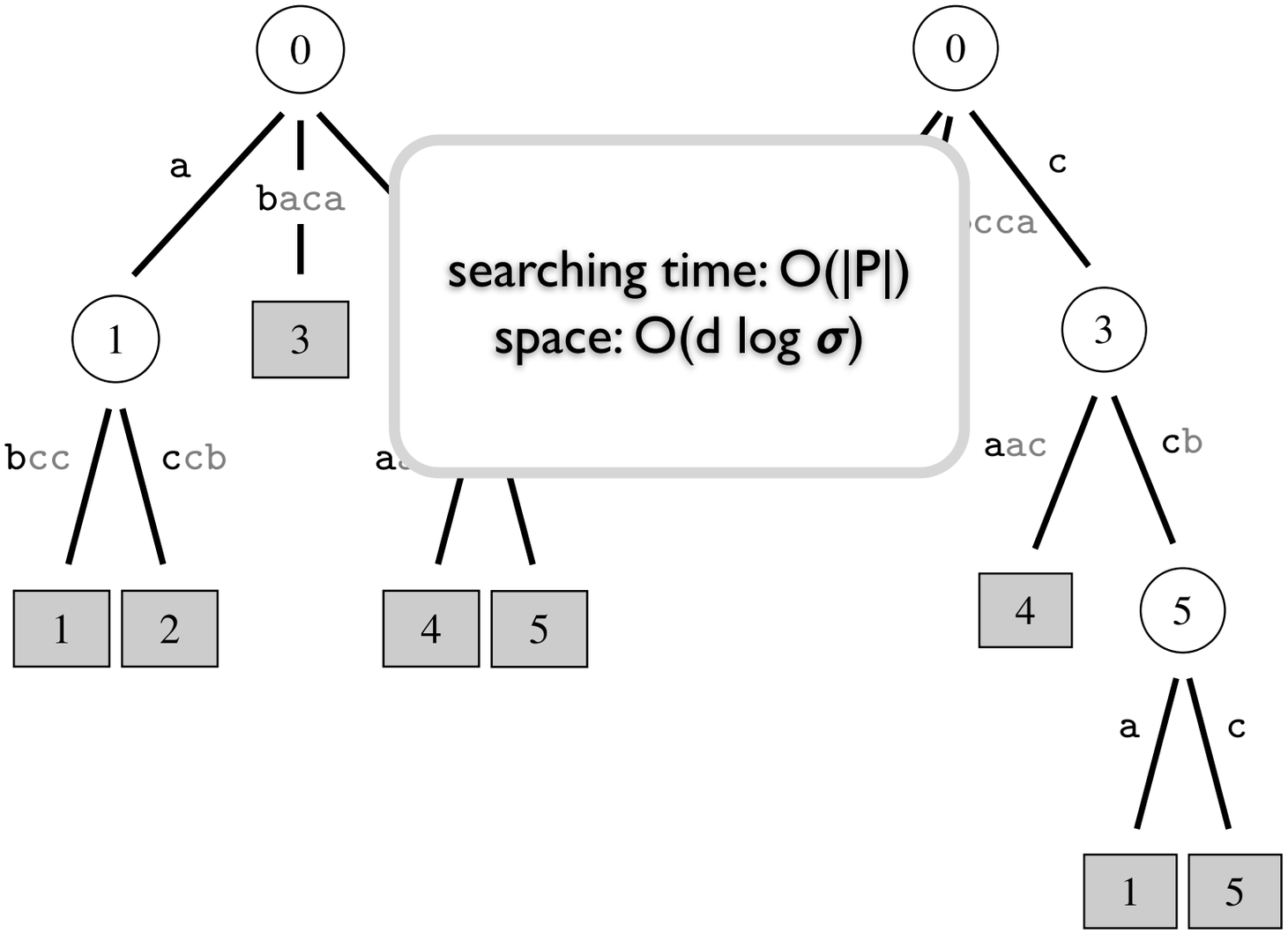
$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc accb baca caac cbcc}$$



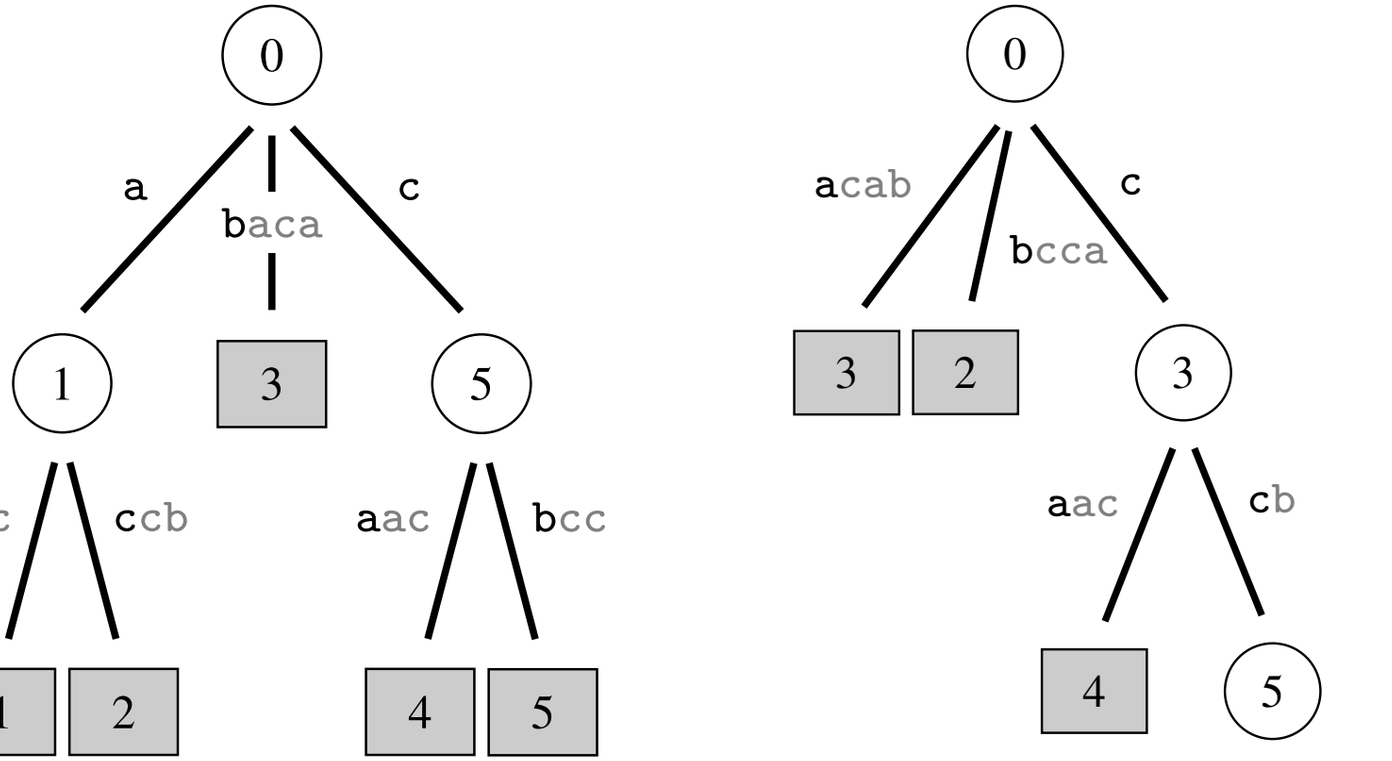
$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc accb baca caac cbcc}$$



$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

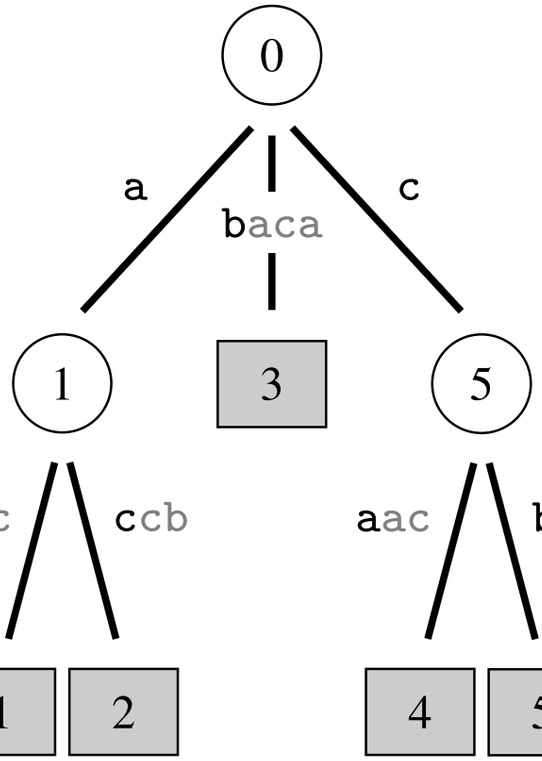
$$cmp(D) = \texttt{abcc accb baca caac cbcc}$$

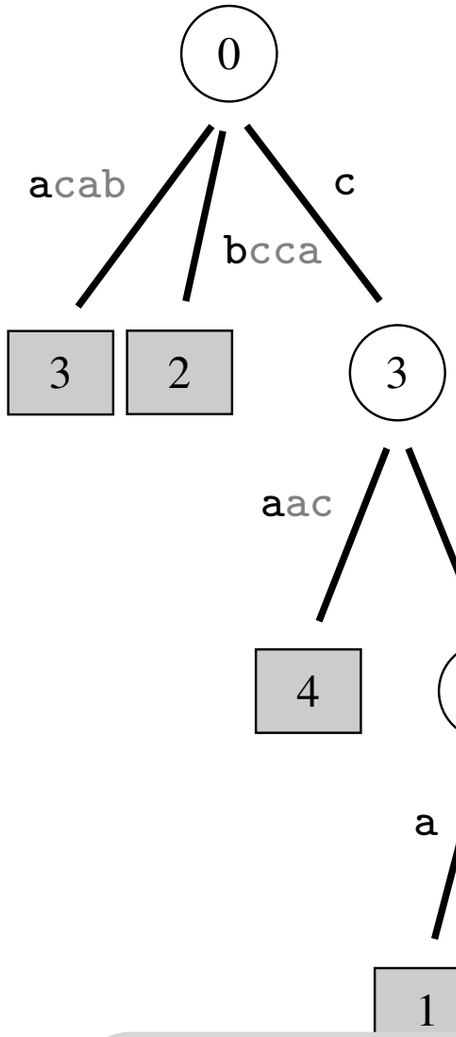


	1	2	3	4	5
lcp	1	0	0	1	×
lcp_r	0	0	1	3	×
π	3	2	4	1	5

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc accb baca caac cbcc}$$



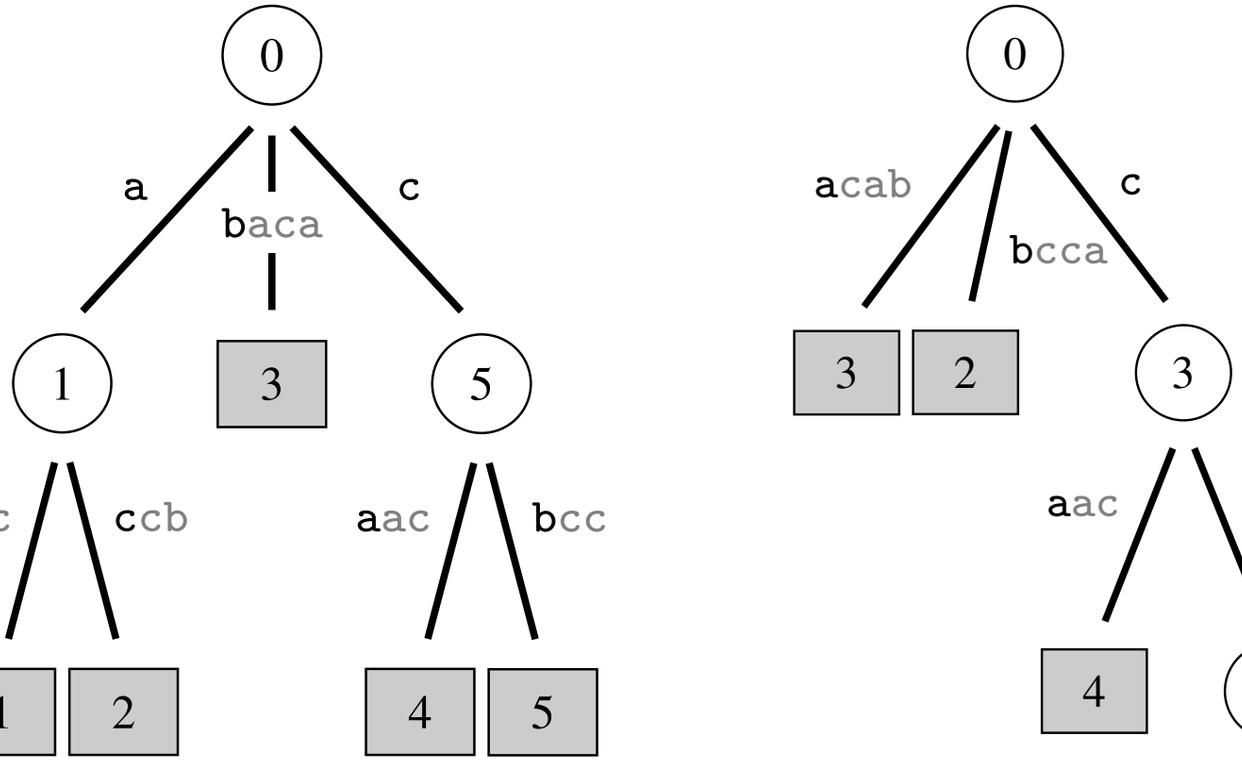


	1	2	3	4	5
lcp	1	0	0	1	×
lcp_r	0	0	1	3	×
π	3	2	4	1	5

any LCP: O(1)
space:
O(d (log d + log (n/d))

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc accb baca caac cbcc}$$



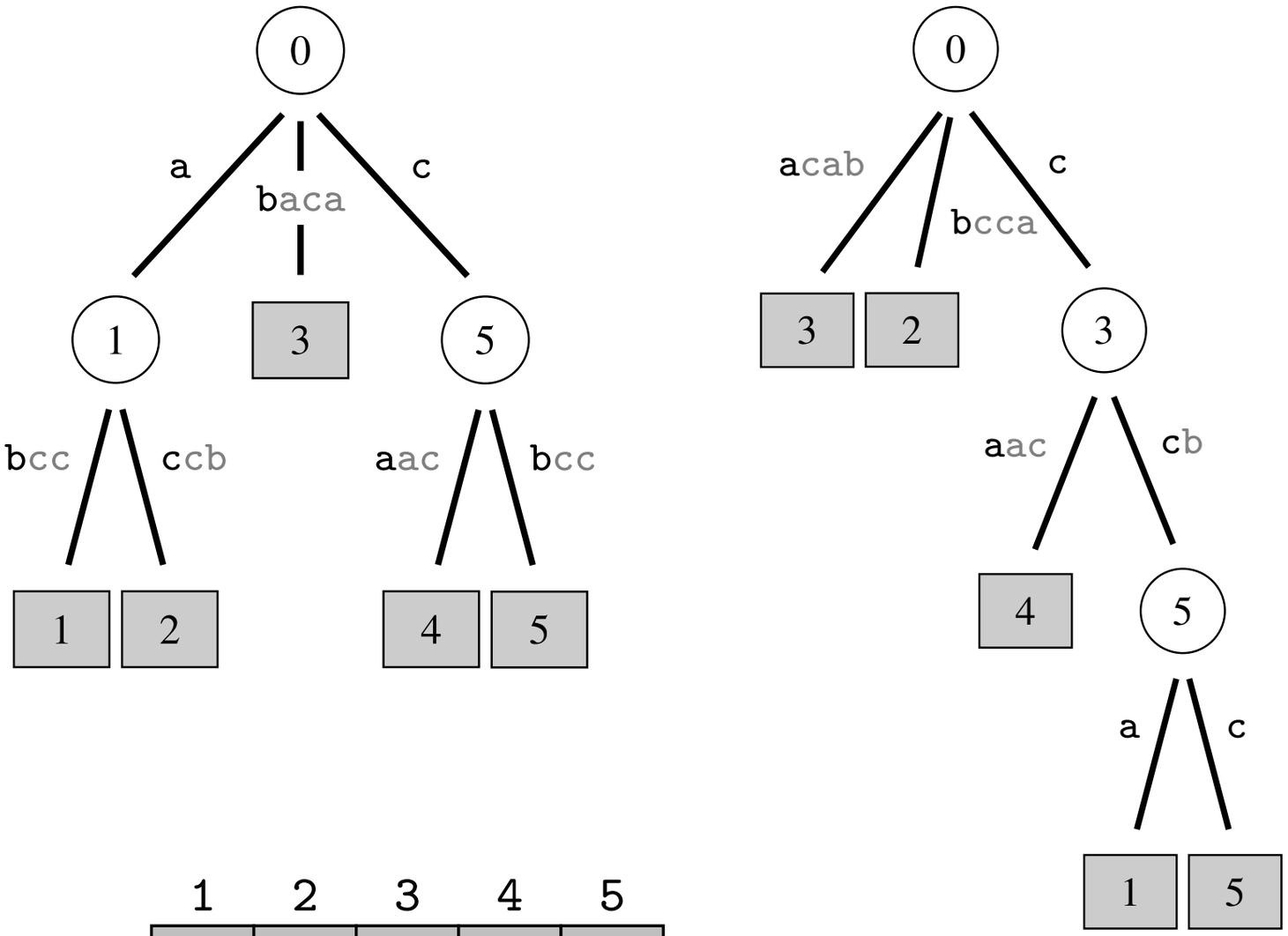
	1	2	3	4	5
lcp	1	0	0	1	×
lcp_r	0	0	1	3	×
π	3	2	4	1	5

Factorizations
of the form

S[1,s] = S[1,i-1] S[i] S[i+1, s]

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc accb baca caac cbcc}$$



	1	2	3	4	5
lcp	1	0	0	1	×
lcp_r	0	0	1	3	×
π	3	2	4	1	5

$\langle 0, \quad 0, \quad \texttt{a}, \quad 3, \quad 5 \rangle$

$\langle 1, \quad 1, \quad \texttt{b}, \quad 2, \quad 3 \rangle$

$\langle 1, \quad 2, \quad \texttt{c}, \quad 1, \quad 3 \rangle$

$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0 \rangle$

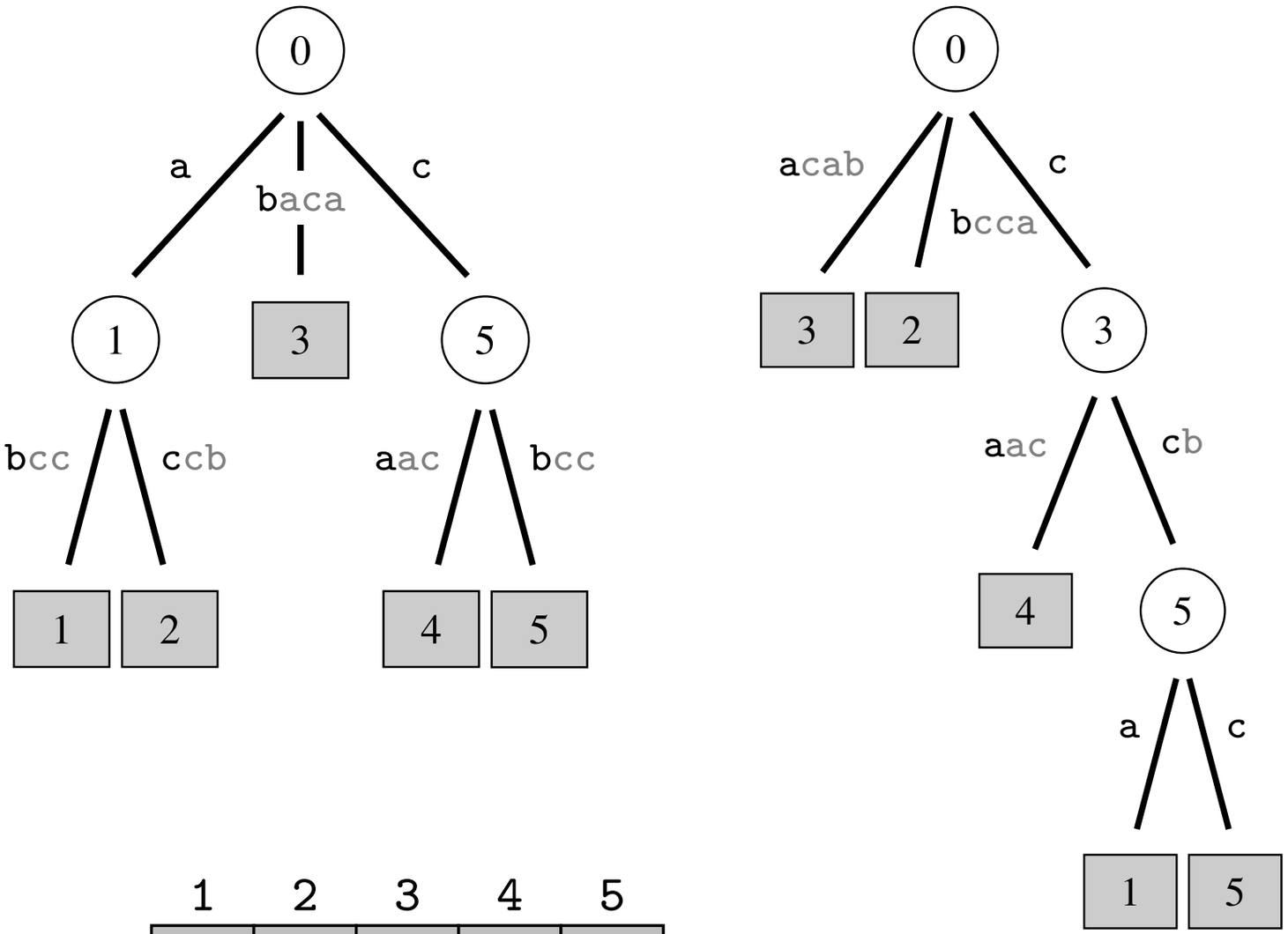
a b c c

Factorizations
of the form

S[1,s] = S[1,i-1] S[i] S[i+1, s]

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc accb baca caac cbcc}$$



	1	2	3	4	5
lcp	1	0	0	1	×
lcp_r	0	0	1	3	×
π	3	2	4	1	5

S[1,i-1] in PT

$\langle 0, \quad 0, \quad \texttt{a}, \quad 3, \quad 5 \rangle$

$\langle 1, \quad 1, \quad \texttt{b}, \quad 2, \quad 3 \rangle$

$\langle 1, \quad 2, \quad \texttt{c}, \quad 1, \quad 3 \rangle$

$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0 \rangle$

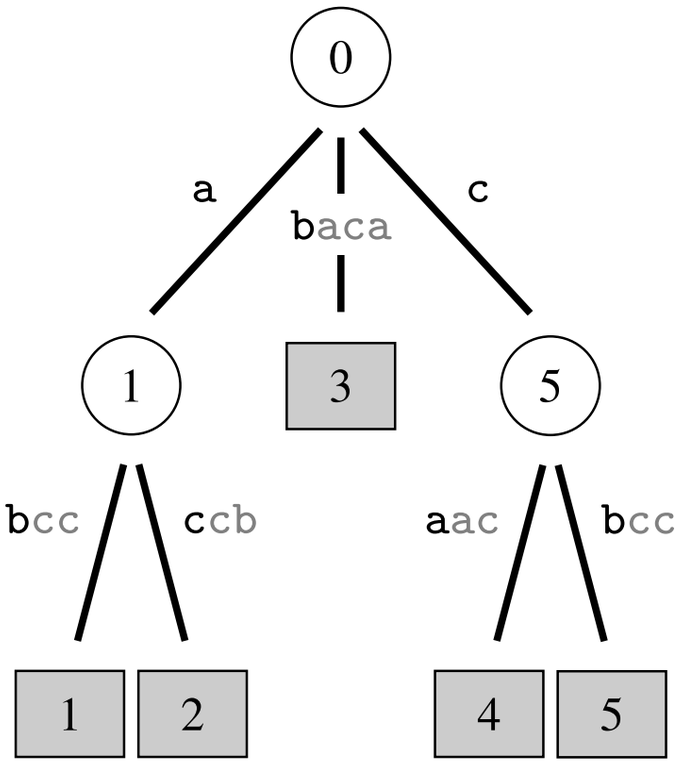
a b c c

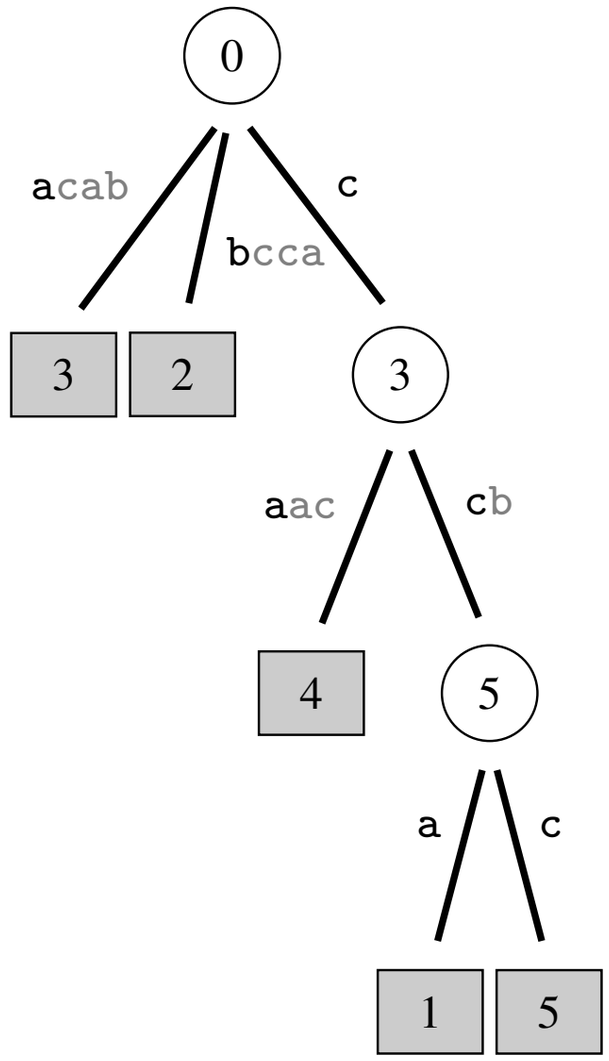
Factorizations
of the form

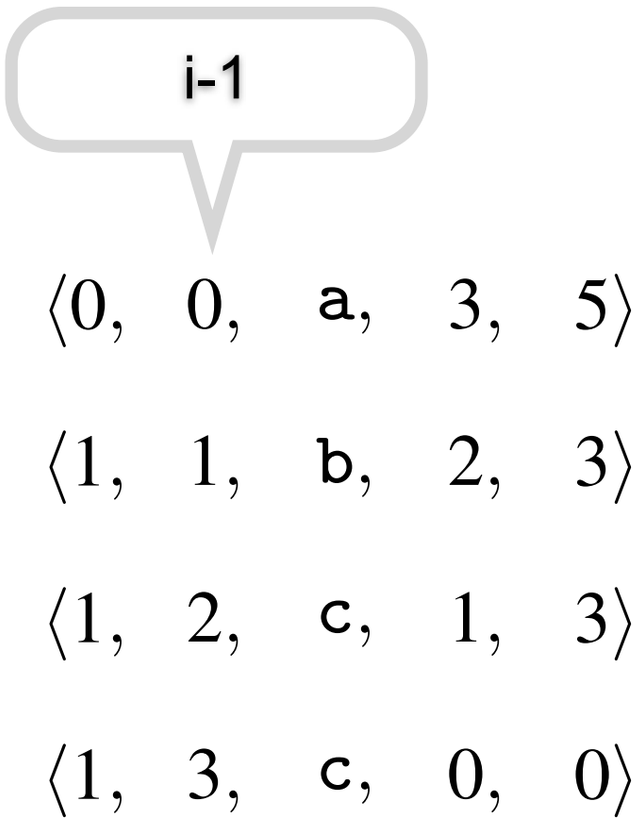
S[1,s] = S[1,i-1] S[i] S[i+1, s]

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc accb baca caac cbcc}$$







a b c c

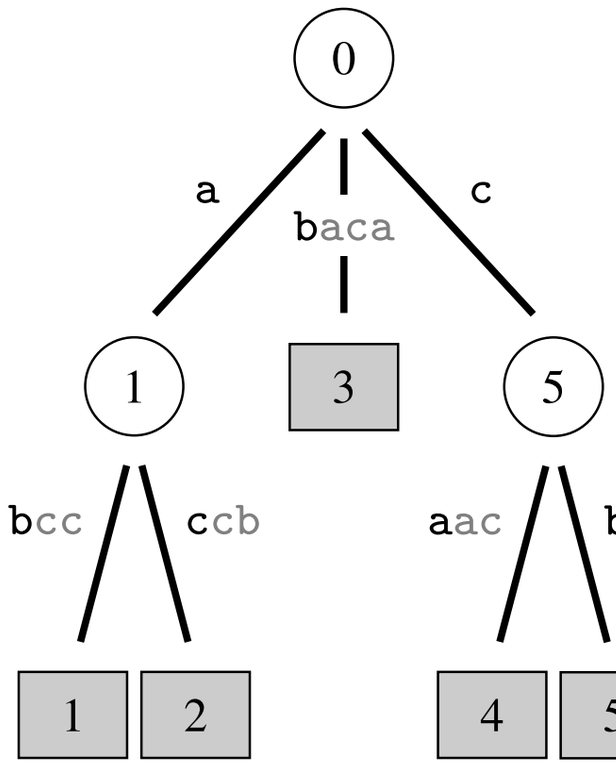
	1	2	3	4	5
lcp	1	0	0	1	×
lcp_r	0	0	1	3	×
π	3	2	4	1	5

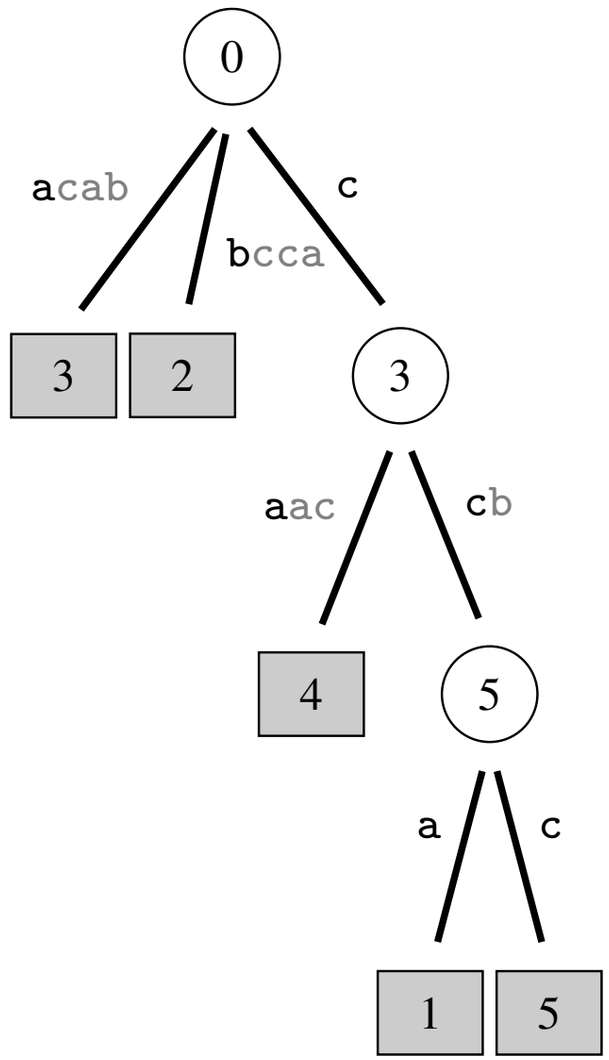
Factorizations
of the form

S[1,s] = S[1,i-1] S[i] S[i+1, s]

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc accb baca caac cbcc}$$





	1	2	3	4	5
lcp	1	0	0	1	×
lcp_r	0	0	1	3	×
π	3	2	4	1	5

S[i]

$\langle 0, \quad 0, \quad \texttt{a}, \quad 3, \quad 5 \rangle$

$\langle 1, \quad 1, \quad \texttt{b}, \quad 2, \quad 3 \rangle$

$\langle 1, \quad 2, \quad \texttt{c}, \quad 1, \quad 3 \rangle$

$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0 \rangle$

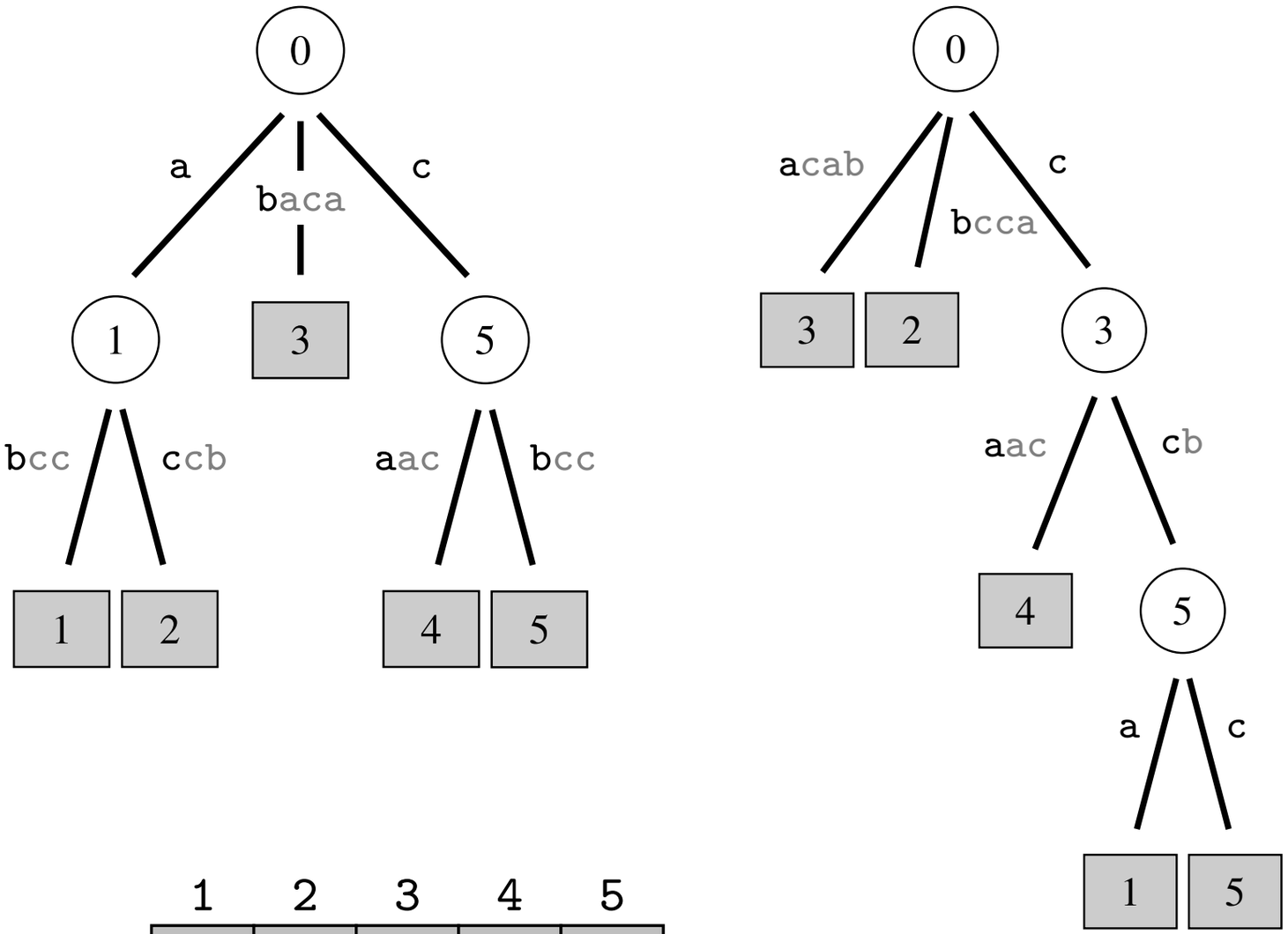
a b c c

Factorizations
of the form

S[1,s] = S[1,i-1] S[i] S[i+1, s]

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc accb baca caac cbcc}$$



	1	2	3	4	5
lcp	1	0	0	1	×
lcp_r	0	0	1	3	×
π	3	2	4	1	5

S[i+1,s] in PT^r

$\langle 0, \quad 0, \quad \texttt{a}, \quad 3, \quad 5 \rangle$

$\langle 1, \quad 1, \quad \texttt{b}, \quad 2, \quad 3 \rangle$

$\langle 1, \quad 2, \quad \texttt{c}, \quad 1, \quad 3 \rangle$

$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0 \rangle$

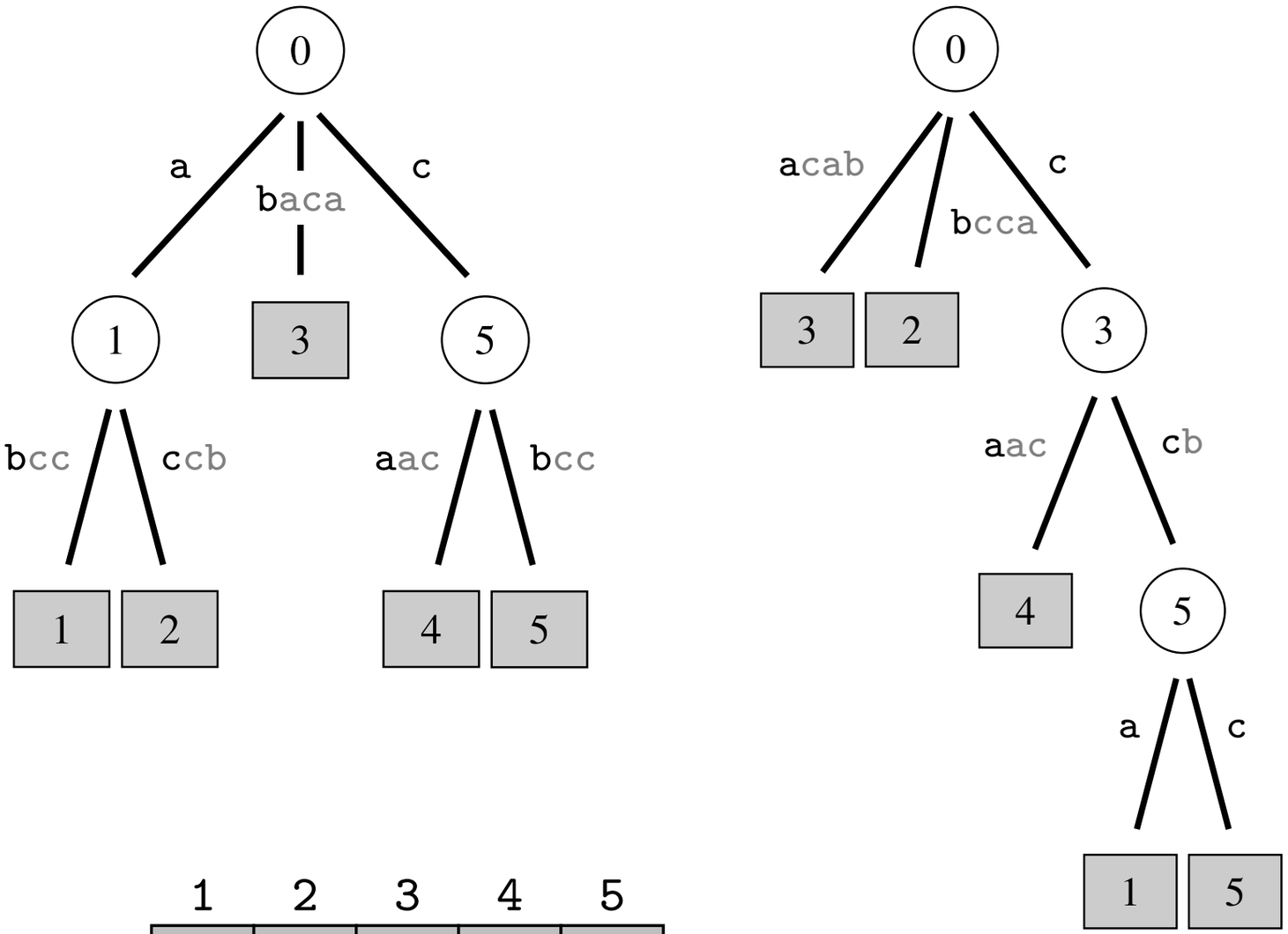
a b c c

Factorizations of the form

S[1,s] = S[1,i-1] S[i] S[i+1, s]

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc accb baca caac cbcc}$$



	1	2	3	4	5
lcp	1	0	0	1	×
lcp_r	0	0	1	3	×
π	3	2	4	1	5

s-i-1

$\langle 0, \quad 0, \quad \texttt{a}, \quad 3, \quad 5 \rangle$

$\langle 1, \quad 1, \quad \texttt{b}, \quad 2, \quad 3 \rangle$

$\langle 1, \quad 2, \quad \texttt{c}, \quad 1, \quad 3 \rangle$

$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0 \rangle$

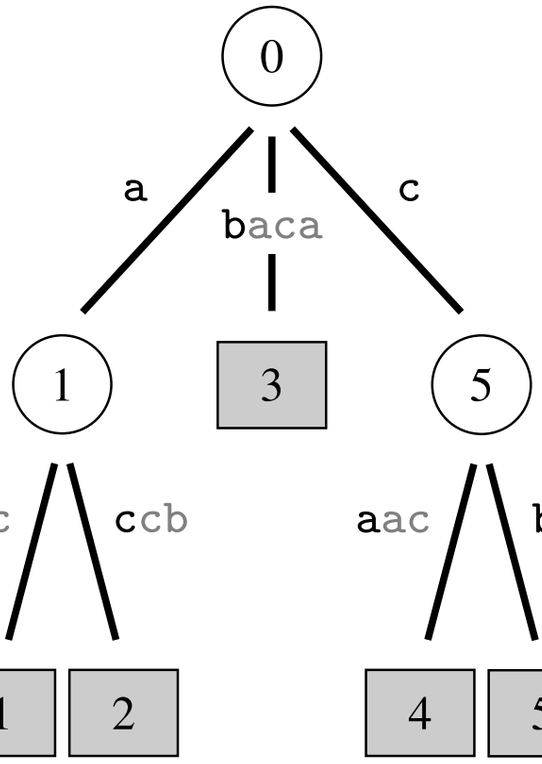
a b c c

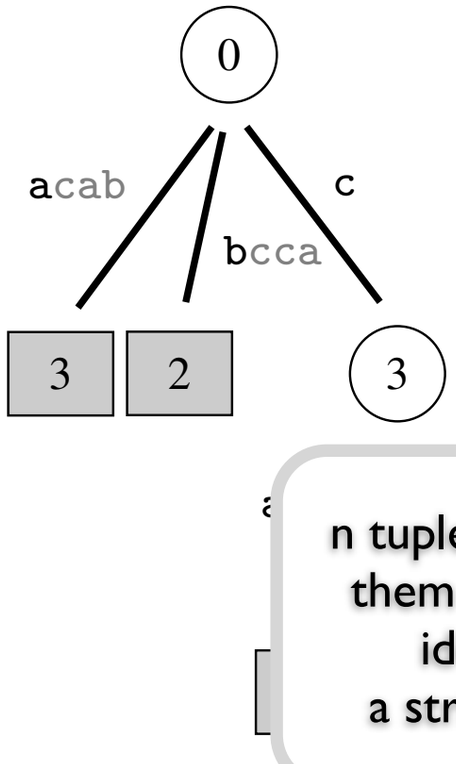
Factorizations
of the form

S[1,s] = S[1,i-1] S[i] S[i+1, s]

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc accb baca caac cbcc}$$





n tuples. Each of them uniquely identify a string in D

$$\langle 0, \quad 0, \quad \texttt{a}, \quad 3, \quad 5 \rangle$$

$$\langle 1, \quad 1, \quad \texttt{b}, \quad 2, \quad 3 \rangle$$

$$\langle 1, \quad 2, \quad \texttt{c}, \quad 1, \quad 3 \rangle$$

$$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0 \rangle$$

a b c c

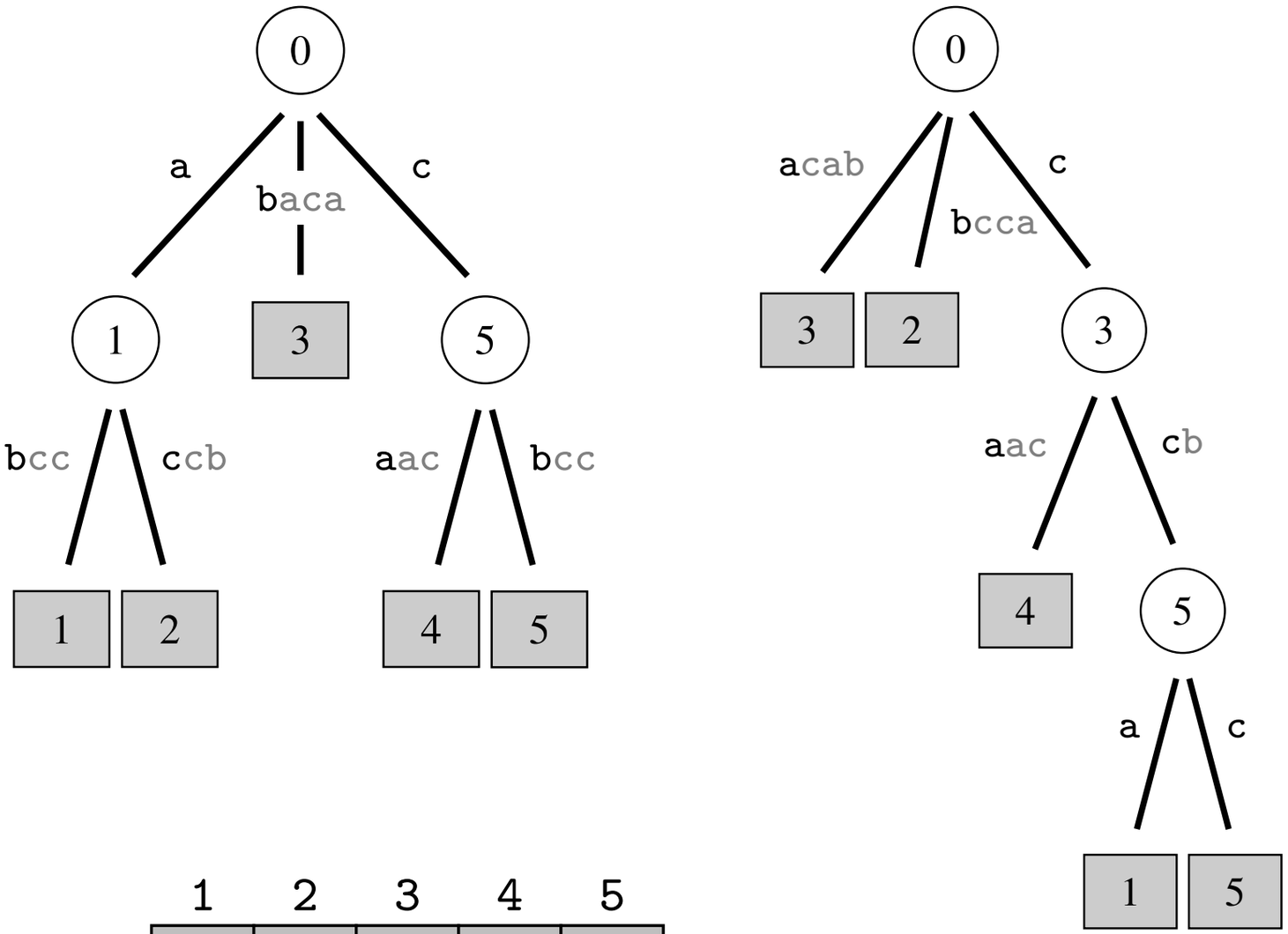
	1	2	3	4	5
lcp	1	0	0	1	×
lcp_r	0	0	1	3	×
π	3	2	4	1	5

Factorizations of the form

S[1,s] = S[1,i-1] S[i] S[i+1, s]

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc accb baca caac cbcc}$$



$\langle 0, \quad 0, \quad \texttt{a}, \quad 3, \quad 5\rangle$

$\langle 1, \quad 1, \quad \texttt{b}, \quad 2, \quad 3\rangle$

$\langle 1, \quad 2, \quad \texttt{c}, \quad 1, \quad 3\rangle$

$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0\rangle$

	1	2	3	4	5
lcp	1	0	0	1	×
lcp_r	0	0	1	3	×
π	3	2	4	1	5

a b c c

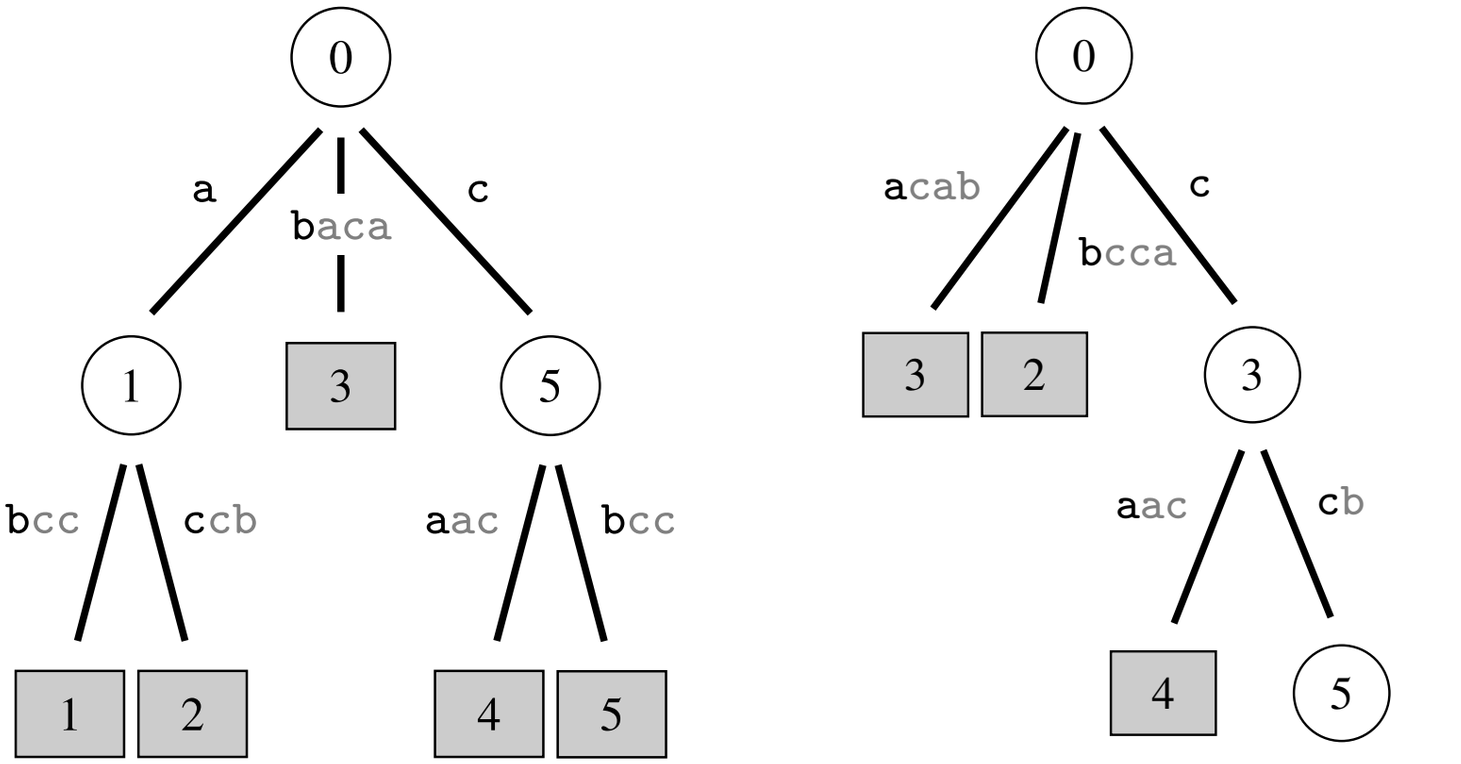
$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc accb baca caac cbcc}$$

Find insertions in P[1,p]

Search factorizations P[1,i] P[i+1,p]

np_i, i sp_i, p-i+1



$\langle 0, \quad 0, \quad \texttt{a}, \quad 3, \quad 5 \rangle$

$\langle 1, \quad 1, \quad \texttt{b}, \quad 2, \quad 3 \rangle$

$\langle 1, \quad 2, \quad \texttt{c}, \quad 1, \quad 3 \rangle$

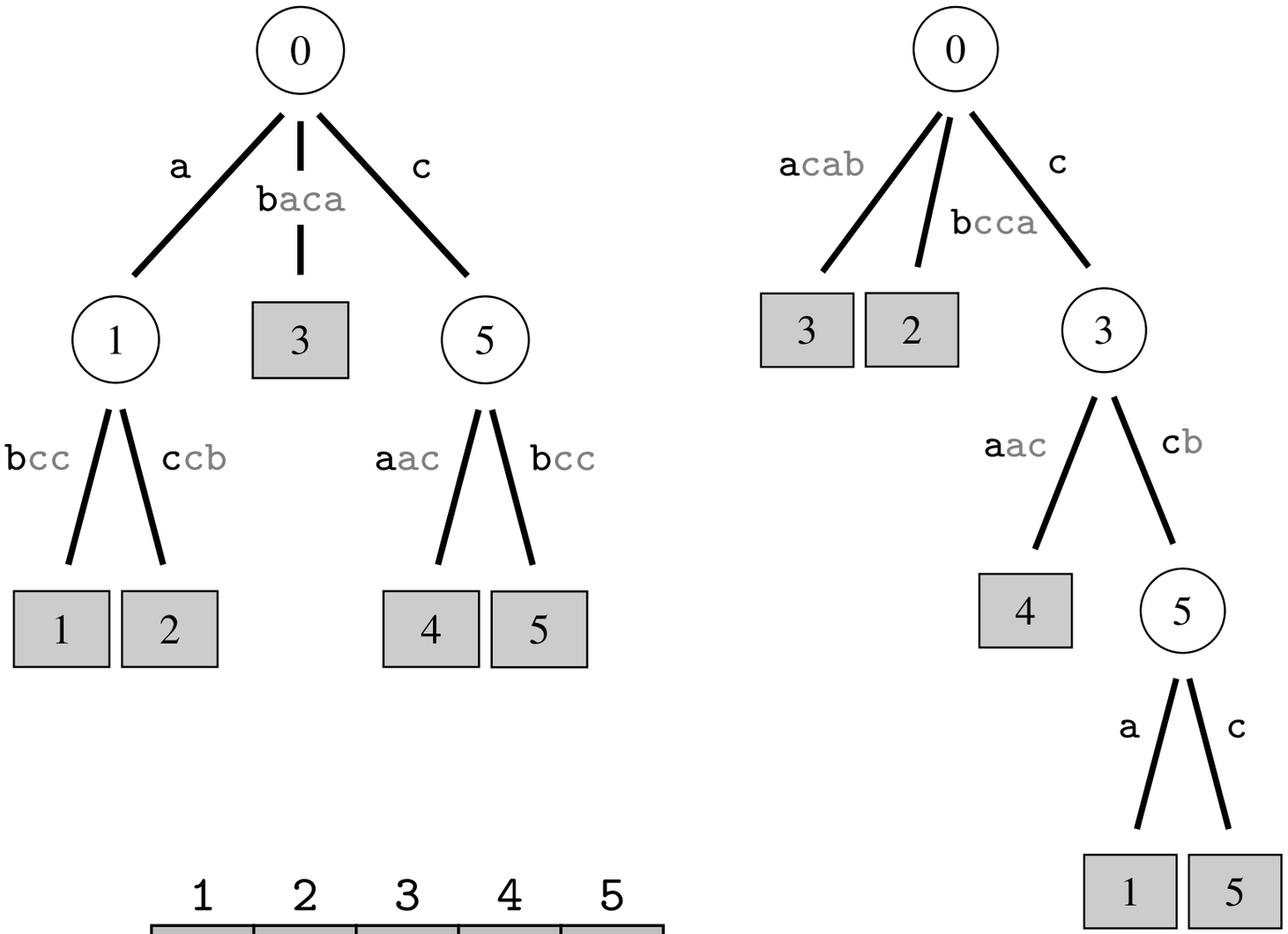
$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0 \rangle$

a b c c

	1	2	3	4	5
lcp	1	0	0	1	×
lcp_r	0	0	1	3	×
π	3	2	4	1	5

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc accb baca caac cbcc}$$



	1	2	3	4	5
lcp	1	0	0	1	×
lcp_r	0	0	1	3	×
π	3	2	4	1	5

Find insertions in P[1,p]

Search factorizations P[1,i] P[i+1,p]

np_i, i sp_i, p-i+1

All in O(p) time

$\langle 0, \quad 0, \quad \texttt{a}, \quad 3, \quad 5 \rangle$

$\langle 1, \quad 1, \quad \texttt{b}, \quad 2, \quad 3 \rangle$

$\langle 1, \quad 2, \quad \texttt{c}, \quad 1, \quad 3 \rangle$

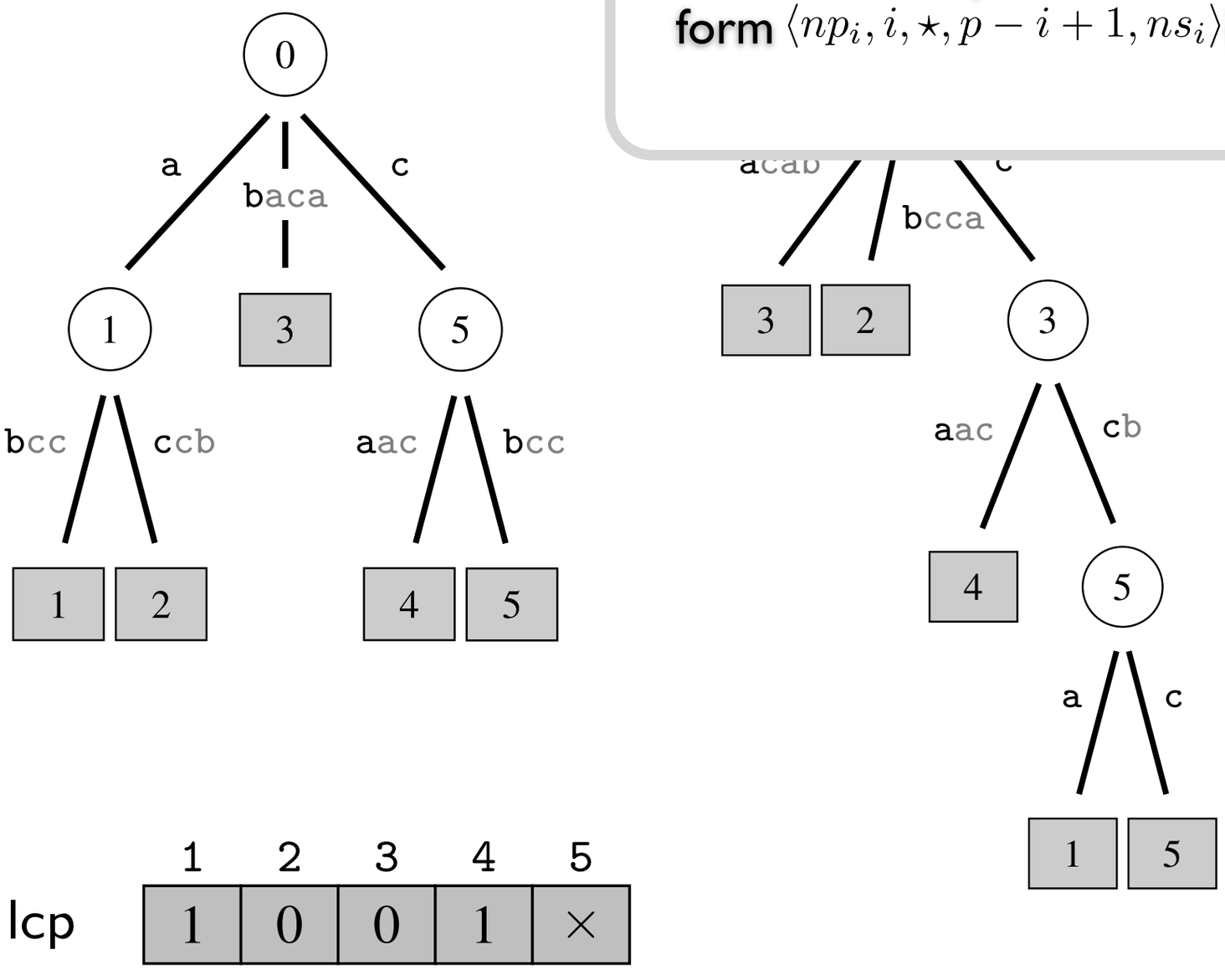
$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0 \rangle$

a b c c

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc ac}$$

Are there tuples of the form $\langle np_i, i, \star, p - i + 1, ns_i \rangle$?



	1	2	3	4	5
lcp	1	0	0	1	×
lcp_r	0	0	1	3	×
π	3	2	4	1	5

Find insertions in P[1,p]

Search factorizations P[1,i] P[i+1,p]

np_i, i sp_i, p-i+1

All in O(p) time

$\langle 0, \quad 0, \quad \texttt{a}, \quad 3, \quad 5 \rangle$

$\langle 1, \quad 1, \quad \texttt{b}, \quad 2, \quad 3 \rangle$

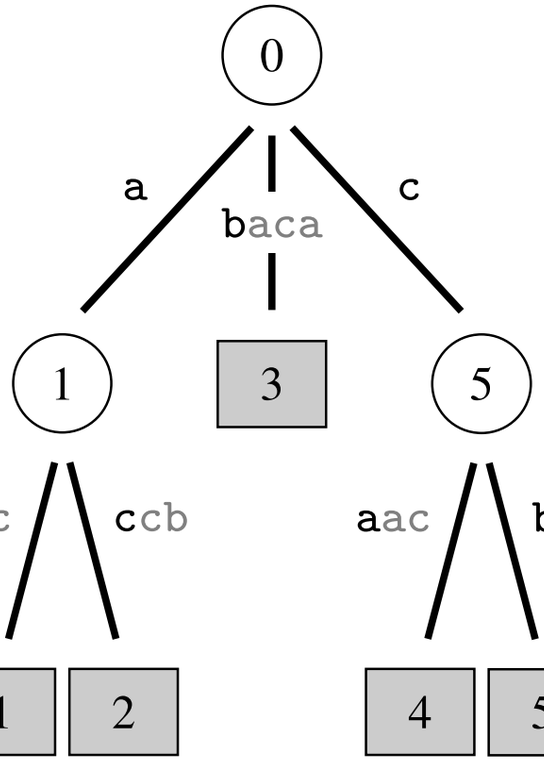
$\langle 1, \quad 2, \quad \texttt{c}, \quad 1, \quad 3 \rangle$

$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0 \rangle$

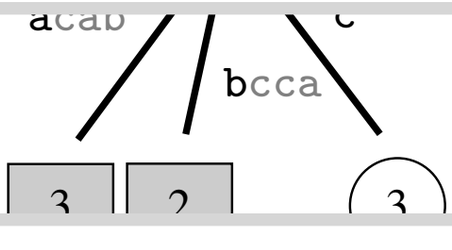
a b c c

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc ac}$$



Are there tuples of the form $\langle np_i, i, \star, p-i+1, ns_i \rangle$?

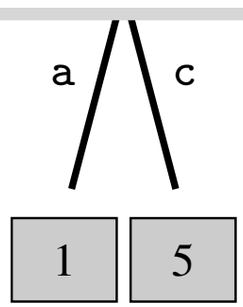


Yes?

$P[1,i] \star P[i+1,p]$

is in D

(and has distance one from P)



Find insertions in P[1,p]

Search factorizations P[1,i] P[i+1,p]

np_i, i $\qquad$ sp_i, p-i+1

All in O(p) time

$\langle 0, \quad 0, \quad \texttt{a}, \quad 3, \quad 5 \rangle$

$\langle 1, \quad 1, \quad \texttt{b}, \quad 2, \quad 3 \rangle$

$\langle 1, \quad 2, \quad \texttt{c}, \quad 1, \quad 3 \rangle$

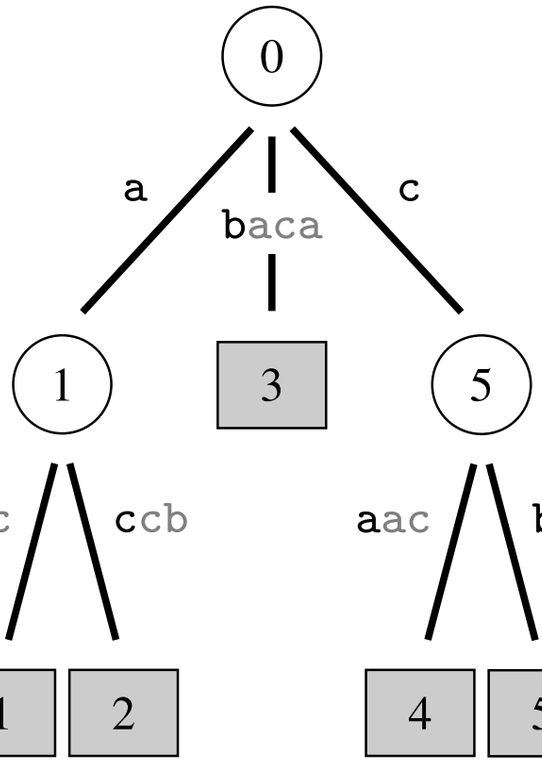
$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0 \rangle$

a b c c

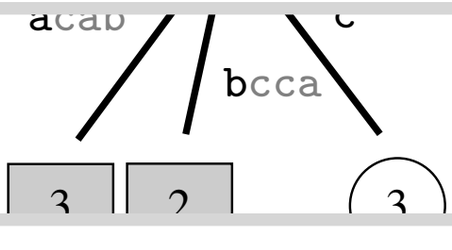
	1	2	3	4	5
lcp	1	0	0	1	×
lcp_r	0	0	1	3	×
π	3	2	4	1	5

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc ac}$$



Are there tuples of the form $\langle np_i, i, \star, p-i+1, ns_i \rangle$?

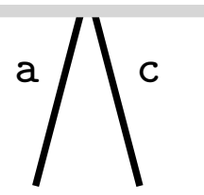


Yes?

$P[1,i] \star P[i+1,p]$

is in D

(and has distance one from P)



Find insertions in P[1,p]

Search factorizations P[1,i] P[i+1,p]

np_i, i $\qquad$ sp_i, p-i+1

All in O(p) time

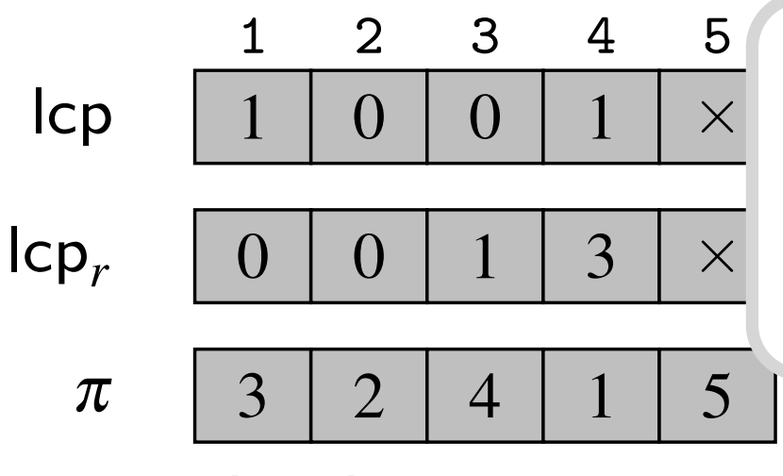
$\langle 0, \quad 0, \quad \texttt{a}, \quad 3, \quad 5 \rangle$

$\langle 1, \quad 1, \quad \texttt{b}, \quad 2, \quad 3 \rangle$

$\langle 1, \quad 2, \quad \texttt{c}, \quad 1, \quad 3 \rangle$

$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0 \rangle$

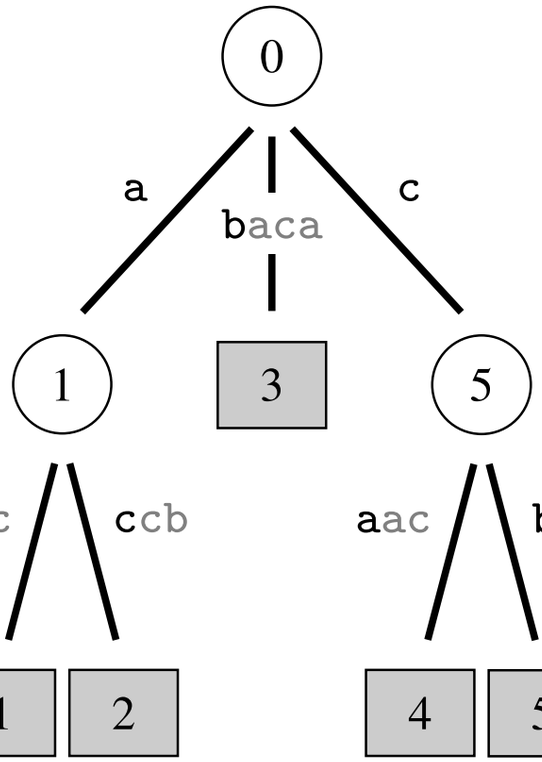
a b c c



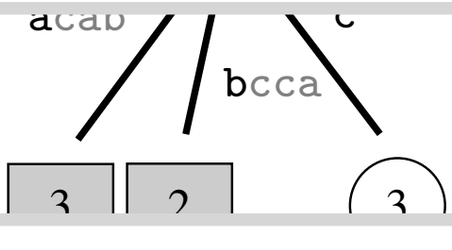
Index tuples so that, given $np_i, i, sp_i, p-i+1$, return the symbols, if any, in tuples of the form $\langle np_i, i, \star, p-i+1, ns_i \rangle$.

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$cmp(D) = \texttt{abcc ac}$$



Are there tuples of the form $\langle np_i, i, \star, p-i+1, ns_i \rangle$?

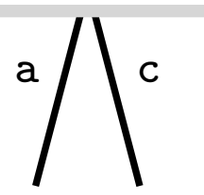


Yes?

$P[1,i] \star P[i+1,p]$

is in D

(and has distance one from P)



Find insertions in P[1,p]

Search factorizations P[1,i] P[i+1,p]

np_i, i $\qquad$ sp_i, p-i+1

All in O(p) time

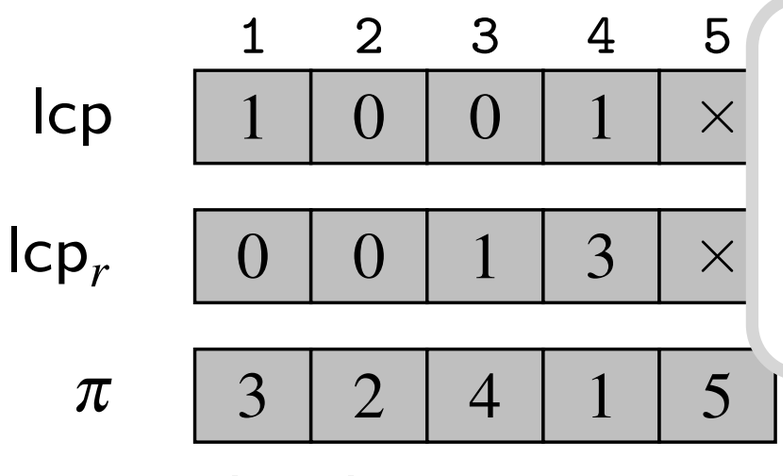
$\langle 0, \quad 0, \quad \texttt{a}, \quad 3, \quad 5 \rangle$

$\langle 1, \quad 1, \quad \texttt{b}, \quad 2, \quad 3 \rangle$

$\langle 1, \quad 2, \quad \texttt{c}, \quad 1, \quad 3 \rangle$

$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0 \rangle$

a b c c



Index tuples so that, given $np_i, i, sp_i, p-i+1$, return the symbols, if any, in tuples of the form $\langle np_i, i, \star, p-i+1, ns_i \rangle$.

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$\langle 0, \quad 0, \quad \texttt{a}, \quad 3, \quad 5 \rangle$$

$$\langle 1, \quad 1, \quad \texttt{b}, \quad 2, \quad 3 \rangle$$

$$\langle 1, \quad 2, \quad \texttt{c}, \quad 1, \quad 3 \rangle$$

$$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0 \rangle$$

Index tuples so that, given $np_i, i, sp_i, p - i + 1$, return the symbols, if any, in tuples of the form $\langle np_i, i, \star, p - i + 1, ns_i \rangle$.

a b c c

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Define the function F, so that,

$$F(np, i, j, sp) = c$$

iff there exists tuple

$$\langle np, i, \mathbf{c}, j, sp \rangle$$

$\langle 0, \quad 0, \quad \texttt{a}, \quad 3, \quad 5 \rangle$

$\langle 1, \quad 1, \quad \texttt{b}, \quad 2, \quad 3 \rangle$

$\langle 1, \quad 2, \quad \texttt{c}, \quad 1, \quad 3 \rangle$

$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0 \rangle$

Index tuples so that,
given $np_i, i, sp_i, p - i + 1$,
return the symbols,
if any, in tuples of the
form $\langle np_i, i, \star, p - i + 1, ns_i \rangle$.

a b c c

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Define the function F, so that,

$$F(np, i, j, sp) = c$$

iff there exists tuple

$$\langle np, i, \mathbf{c}, j, sp \rangle$$

A function F from a subset S ⊆ U of size n
to an alphabet of size σ can be represented within
$nH_0 + n\, o(H_0)$ bits, so that
if x ∈ S, F(x) evaluated in constant time
if x ∉ S, an arbitrary symbol is returned

$$\langle 0, \quad 0, \quad \texttt{a}, \quad 3, \quad 5 \rangle$$

$$\langle 1, \quad 1, \quad \texttt{b}, \quad 2, \quad 3 \rangle$$

$$\langle 1, \quad 2, \quad \texttt{c}, \quad 1, \quad 3 \rangle$$

$$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0 \rangle$$

a b c c

Index tuples so that,
given $np_i, i, sp_i, p - i + 1$,
return the symbols,
if any, in tuples of the
form $\langle np_i, i, \star, p - i + 1, ns_i \rangle$.

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Define the function F, so that,

$$F(np, i, j, sp) = c$$

iff there exists tuple

$$\langle np, i, \mathbf{c}, j, sp \rangle$$

A function F from a subset S ⊆ U of size n to an alphabet of size σ can be represented within nH₀ + n o(H₀) bits, so that if x ∈ S, F(x) evaluated in constant time if x ∉ S, an arbitrary symbol is returned

H_k is achieved by splitting F into several functions based on k-long context preceding the symbol

$$\langle 0, \quad 0, \quad \texttt{a}, \quad 3, \quad 5 \rangle$$

$$\texttt{b}, \quad 2, \quad 3 \rangle$$

$$\texttt{c}, \quad 1, \quad 3 \rangle$$

$$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0 \rangle$$

a b c c

Index tuples so that, given $np_i, i, sp_i, p - i + 1$, return the symbols, if any, in tuples of the form $\langle np_i, i, \star, p - i + 1, ns_i \rangle$.

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Define the function F, so that,

$$F(np, i, j, sp) = c$$

iff there exists tuple

$$\langle np, i, \mathbf{c}, j, sp \rangle$$

A function F from a subset S ⊆ U of size n
to an alphabet of size σ can be represented within
$nH_0 + n\, o(H_0)$ bits, so that
if x ∈ S, F(x) evaluated in constant time
if x ∉ S, an arbitrary symbol is returned

$\langle 0, \quad 0, \quad \texttt{a}, \quad 3, \quad 5 \rangle$

$\langle 1, \quad 1, \quad \texttt{b}, \quad 2, \quad 3 \rangle$

$\langle 1, \quad 2, \quad \texttt{c}, \quad 1, \quad 3 \rangle$

$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0 \rangle$

a b c c

Index tuples so that,
given $np_i, i, sp_i, p - i + 1$,
return the symbols,
if any, in tuples of the
form $\langle np_i, i, \star, p - i + 1, ns_i \rangle$.

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Define the function F, so that,

$$F(np, i, j, sp) = c$$

iff there exists tuple

$$\langle np, i, \mathbf{c}, j, sp \rangle$$

A function F from a subset S ⊆ U of size n
to an alphabet of size σ can be represented within
$nH_0 + n\, o(H_0)$ bits, so that
if x ∈ S, F(x) evaluated in constant time
if x ∉ S, an arbitrary symbol is returned

We may err!
P[1,i] c P[i+1,p]
may not be in D.

We need to check
if it belongs to D!

tuples so that,
$p_i, i, sp_i, p - i + 1,$
n the symbols,
n tuples of the
$, \star, p - i + 1, ns_i \rangle$.

$\langle 0, \quad 0, \quad \texttt{a}, \quad 3, \quad 5 \rangle$

$\langle 1, \quad 1, \quad \texttt{b}, \quad 2, \quad 3 \rangle$

$\langle 1, \quad 2, \quad \texttt{c}, \quad 1, \quad 3 \rangle$

$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0 \rangle$

a b c c

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Define the function F, so that,

$$F(np, i, j, sp) = c$$

iff there exists tuple

$$\langle np, i, \mathbf{c}, j, sp \rangle$$

A function F from a subset S ⊆ U of size n
to an alphabet of size σ can be represented within
$nH_0 + n\, o(H_0)$ bits, so that
if $x \in S$, F(x) evaluated in constant time
if $x \notin S$, an arbitrary symbol is returned

We may err!
P[1,i] c P[i+1,p]
may not be in D.

We need to check
if it belongs to D

tuples so that,
$p_i, i, sp_i, p - i + 1,$
the symbols,
the
$ns_i\rangle$.

in O(1).
see the paper

$\langle 0, \quad 0, \quad \texttt{a}, \quad 3, \quad 5 \rangle$

$\langle 1, \quad 1, \quad \texttt{b}, \quad 2, \quad 3 \rangle$

$\langle 1, \quad 2, \quad \texttt{c}, \quad 1, \quad 3 \rangle$

$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0 \rangle$

a b c c

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Define the function F, so that,

$$F(np, i, j, sp) = c$$

iff there exists tuple

$$\langle np, i, \mathbf{c}, j, sp \rangle$$

Space: $2nH_k + n \cdot o(\log \sigma) + 2d \log d$

Time: $O(|P| + occ)$

We may err!
P[1,i] c P[i+1,p]
may not be in D.

We need to check
if it belongs to D

tuples so that,

$p_i, i, sp_i, p - i + 1,$

the symbols,

the

$ns_i\rangle$.

in O(1).
see the paper

$$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0 \rangle$$

a b c c

$D = \{\texttt{abcc}, \texttt{ac}$ ... $\texttt{cbcc}\}$

Define the fun

$F(np, i, j$

iff there e

$\langle np, i,$

Compressed Representation + Compressed function

LCP + LCS + identifiers

Space: $2nH_k + n \cdot o(\log \sigma) + 2d \log d$

Time: $O(|P| + occ)$

We may err!
P[1,i] c P[i+1,p]
may not be in D.

We need to check
if it belongs to D

tuples so that,

$p_i, i, sp_i, p - i + 1,$

the symbols,

the

$ns_i\rangle$.

in O(1).
see the paper

$\langle 1, \quad 3, \quad \texttt{c}, \quad 0, \quad 0\rangle$

a b c c

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Subset of strings having the same length p+1

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Subset of strings having the same length p+1

a b c c #
b c c # a
c c # a b
c # a b c
a b c c

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

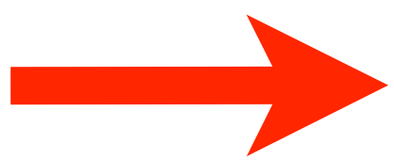
Subset of strings having the same length p+1

a b c c #
b c c # a
c c # a b
c # a b c
a b c c
a c c b #
c c b # a
c b # a c
b # a c c
a c c b
b a c a #
a c a # b
c a # b a
a # b a c
b a c a
c a a c #
a a c # c
a c # c a
c # c a a
c a a c
c b c c #
b c c # c
c c # c b
c # c b c
c b c c

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Subset of strings having the same length p+1

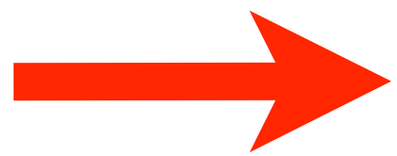
a b c c #
b c c # a
c c # a b
c # a b c
a b c c
a c c b #
c c b # a
c b # a c
b # a c c
a c c b
b a c a #
a c a # b
c a # b a
a # b a c
b a c a
c a a c #
a a c # c
a c # c a
c # c a a
c a a c
c b c c #
b c c # c
c c # c b
c # c b c
c b c c



$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Subset of strings having the same length p+1

```
a b c c #
b c c # a
c c # a b
c # a b c
# a b c c
a c c b #
c c b # a
c b # a c
b # a c c
# a c c b
b a c a #
a c a # b
c a # b a
a # b a c
# b a c a
c a a c #
a a c # c
a c # c a
c # c a a
# c a a c
c b c c #
b c c # c
c c # c b
c # c b c
# c b c c
```



F				L
#	a	b	c	c
#	a	c	c	b
#	b	a	c	a
#	c	a	a	c
#	c	b	c	c
a	#	b	a	c
a	a	c	#	c
a	b	c	c	#
a	c	#	c	a
a	c	a	#	b
a	c	c	b	#
b	#	a	c	c
b	a	c	a	#
b	c	c	#	a
b	c	c	#	c
c	#	a	b	c
c	#	c	a	a
c	#	c	b	c
c	a	#	b	a
c	a	a	c	#
c	b	#	a	c
c	b	c	c	#
c	c	#	a	b
c	c	#	c	b
c	c	b	#	a

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Subset of strings having the same length p+1

F				L
#	a	b	c	c
#	a	c	c	b
#	b	a	c	a
#	c	a	a	c
#	c	b	c	c
a	#	b	a	c
a	a	c	#	c
a	b	c	c	#
a	c	#	c	a
a	c	a	#	b
a	c	c	b	#
b	#	a	c	c
b	a	c	a	#
b	c	c	#	a
b	c	c	#	c
c	#	a	b	c
c	#	c	a	a
c	#	c	b	c
c	a	#	b	a
c	a	a	c	#
c	b	#	a	c
c	b	c	c	#
c	c	#	a	b
c	c	#	c	b
c	c	b	#	a

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Subset of strings having the same length p+1

It preserves nice properties of Burrows-Wheeler Transform

F				L
#	a	b	c	c
a	c	c	b	b
b	a	c	a	a
c	a	a	c	c
#	a	b	c	c
a	#	b	a	c
a	a	c	#	c
a	b	c	c	#
a	c	#	c	a
a	c	a	#	b
a	c	c	b	#
b	#	a	c	c
b	a	c	a	#
b	c	c	#	a
b	c	c	#	c
c	#	a	b	c
c	#	c	a	a
c	#	c	b	c
c	a	#	b	a
c	a	a	c	#
c	b	#	a	c
c	b	c	c	#
c	c	#	a	b
c	c	#	c	b
c	c	b	#	a

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Given P[1,p],
identify the rows prefixed by a cyclic rotation of P on the subset with string of len p+1
i.e., P[i,p]#P[1,i-1]

Subset of strings having the same length p+1

F				L
#	a	b	c	c
#	a	c	c	b
#	b	a	c	a
#	c	a	a	c
#	c	b	c	c
a	#	b	a	c
a	a	c	#	c
a	b	c	c	#
a	c	#	c	a
a	c	a	#	b
a	c	c	b	#
b	#	a	c	c
b	a	c	a	#
b	c	c	#	a
b	c	c	#	c
c	#	a	b	c
c	#	c	a	a
c	#	c	b	c
c	a	#	b	a
c	a	a	c	#
c	b	#	a	c
c	b	c	c	#
c	c	#	a	b
c	c	#	c	b
c	c	b	#	a

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Given P[1,p],
identify the rows prefixed by a cyclic rotation of P on the subset with string of len p+1
i.e., P[i,p]#P[1,i-1]

Strings corresponding to these rows are at distance one from P (an insertion)!

Subset of strings having the same length p+1

F				L
#	a	b	c	c
#	a	c	c	b
#	b	a	c	a
#	c	a	a	c
#	c	b	c	c
a	#	b	a	c
a	a	c	#	c
a	b	c	c	#
a	c	#	c	a
a	c	a	#	b
a	c	c	b	#
b	#	a	c	c
b	a	c	a	#
b	c	c	#	a
b	c	c	#	c
c	#	a	b	c
c	#	c	a	a
c	#	c	b	c
c	a	#	b	a
c	a	a	c	#
c	b	#	a	c
c	b	c	c	#
c	c	#	a	b
c	c	#	c	b
c	c	b	#	a

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Given P[1,p],
identify the rows prefixed by a cyclic rotation of P on the subset with string of len p+1
i.e., P[i,p]#P[1,i-1]

Strings corresponding to these rows are at distance one from P (an insertion)!

P = acc

Subset of strings having the same length p+1

F				L
#	a	b	c	c
#	a	c	c	b
#	b	a	c	a
#	c	a	a	c
#	c	b	c	c
a	#	b	a	c
a	a	c	#	c
a	b	c	c	#
a	c	#	c	a
a	c	a	#	b
a	c	c	b	#
b	#	a	c	c
b	a	c	a	#
b	c	c	#	a
b	c	c	#	c
c	#	a	b	c
c	#	c	a	a
c	#	c	b	c
c	a	#	b	a
c	a	a	c	#
c	b	#	a	c
c	b	c	c	#
c	c	#	a	b
c	c	#	c	b
c	c	b	#	a

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Given P[1,p],
identify the rows prefixed by a cyclic rotation of P on the subset with string of len p+1
i.e., P[i,p]#P[1,i-1]

Strings corresponding to these rows are at distance one from P (an insertion)!

P = acc
One rotation is cc#a

Subset of strings having the same length p+1

F				L
#	a	b	c	c
#	a	c	c	b
#	b	a	c	a
#	c	a	a	c
#	c	b	c	c
a	#	b	a	c
a	a	c	#	c
a	b	c	c	#
a	c	#	c	a
a	c	a	#	b
a	c	c	b	#
b	#	a	c	c
b	a	c	a	#
b	c	c	#	a
b	c	c	#	c
c	#	a	b	c
c	#	c	a	a
c	#	c	b	c
c	a	#	b	a
c	a	a	c	#
c	b	#	a	c
c	b	c	c	#
c	c	#	a	b
c	c	#	c	b
c	c	b	#	a

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Given P[1,p],
identify the rows prefixed by a cyclic rotation of P on the subset with string of len p+1
i.e., P[i,p]#P[1,i-1]

Strings corresponding to these rows are at distance one from P (an insertion)!

P = acc
One rotation is cc#a
abcc is in D

Subset of strings having the same length p+1

F				L
#	a	b	c	c
#	a	c	c	b
#	b	a	c	a
#	c	a	a	c
#	c	b	c	c
a	#	b	a	c
a	a	c	#	c
a	b	c	c	#
a	c	#	c	a
a	c	a	#	b
a	c	c	b	#
b	#	a	c	c
b	a	c	a	#
b	c	c	#	a
b	c	c	#	c
c	#	a	b	c
c	#	c	a	a
c	#	c	b	c
c	a	#	b	a
c	a	a	c	#
c	b	#	a	c
c	b	c	c	#
c	c	#	a	b
c	c	#	c	b
c	c	b	#	a

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Given P[1,p],
identify the rows prefixed by a cyclic rotation of P on the subset with string of len p+1
i.e., P[i,p]#P[1,i-1]

Strings corresponding to these rows are at distance one from P (an insertion)!

P = acc
One rotation is cc#a
abcc is in D

How to identify rows prefixed by P[i,p]#P[1,i-1]?

Subset of strings having the same length p+1

F				L
#	a	b	c	c
#	a	c	c	b
#	b	a	c	a
#	c	a	a	c
#	c	b	c	c
a	#	b	a	c
a	a	c	#	c
a	b	c	c	#
a	c	#	c	a
a	c	a	#	b
a	c	c	b	#
b	#	a	c	c
b	a	c	a	#
b	c	c	#	a
b	c	c	#	c
c	#	a	b	c
c	#	c	a	a
c	#	c	b	c
c	a	#	b	a
c	a	a	c	#
c	b	#	a	c
c	b	c	c	#
c	c	#	a	b
c	c	#	c	b
c	c	b	#	a

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Given P[1,p],
identify the rows prefixed by a cyclic rotation of P on the subset with string of len p+1
i.e., P[i,p]#P[1,i-1]

Strings corresponding to these rows are at distance one from P (an insertion)!

P = acc
One rotation is cc#a
abcc is in D

How to identify rows prefixed by P[i,p]#P[1,i-1]?

-) Backward search for each of them? $O(p^2)$ time

Subset of strings having the same length p+1

F				L
#	a	b	c	c
#	a	c	c	b
#	b	a	c	a
#	c	a	a	c
#	c	b	c	c
a	#	b	a	c
a	a	c	#	c
a	b	c	c	#
a	c	#	c	a
a	c	a	#	b
a	c	c	b	#
b	#	a	c	c
b	a	c	a	#
b	c	c	#	a
b	c	c	#	c
c	#	a	b	c
c	#	c	a	a
c	#	c	b	c
c	a	#	b	a
c	a	a	c	#
c	b	#	a	c
c	b	c	c	#
c	c	#	a	b
c	c	#	c	b
c	c	b	#	a

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Given P[1,p],
identify the rows prefixed by a cyclic rotation of P on the subset with string of len p+1
i.e., P[i,p]#P[1,i-1]

Strings corresponding to these rows are at distance one from P (an insertion)!

P = acc
One rotation is cc#a
abcc is in D

How to identify rows prefixed by P[i,p]#P[1,i-1]?

-) Backward search for each of them? $O(p^2)$ time

-) Move from P[i,p]#P[1,i-1] to P[i-1,p]#P[1,i-2] in $O(\log \log \sigma)$

Subset of strings having the same length p+1

F				L
#	a	b	c	c
#	a	c	c	b
#	b	a	c	a
#	c	a	a	c
#	c	b	c	c
a	#	b	a	c
a	a	c	#	c
a	b	c	c	#
a	c	#	c	a
a	c	a	#	b
a	c	c	b	#
b	#	a	c	c
b	a	c	a	#
b	c	c	#	a
b	c	c	#	c
c	#	a	b	c
c	#	c	a	a
c	#	c	b	c
c	a	#	b	a
c	a	a	c	#
c	b	#	a	c
c	b	c	c	#
c	c	#	a	b
c	c	#	c	b
c	c	b	#	a

$$D = \{\texttt{abcc}, \texttt{accb}, \texttt{baca}, \texttt{caac}, \texttt{cbcc}\}$$

Subset of strings having the same length p+1

Given P[1,p],
identify the rows prefixed by a cyclic rotation of P on the subset with string of len p+1
i.e., P[i,p]#P[1,i-1]

Strings corresponding to

F				L
#	a	b	c	c
#	a	c	c	b
#	b	a	c	a
#	c	a	a	c
#	c	b	c	c
a	#	b	a	c
a	a	c	#	c
b	c	c	#	c
c	#	a	b	c
c	#	c	a	a
c	#	c	b	c
c	a	#	b	a
c	a	a	c	#
c	b	#	a	c
c	b	c	c	#
c	c	#	a	b
c	c	#	c	b
c	c	b	#	a

How to identify rows prefixed by P[i,p]#P[1,i-1]?

-) Backward search for each of them? $O(p^2)$ time

-) Move from P[i,p]#P[1,i-1] to P[i-1,p]#P[1,i-2] in O(log log σ)

Space: $nH_k + n \cdot o(\log \sigma)$

Time: $O(|P| \log \log \sigma + occ)$

Open Problems

Open Problems

Solution based on Patricia tries and Perfect

Time $O(|P| + occ)$

Space $2nH_k + n \cdot o(\log \sigma) + 2d \log d$

Top-k strings in $O(|P| + k \log k)$

Open Problems

Solution based on Patricia tries and Perfect

Time $O(|P| + occ)$

Space $2nH_k + n \cdot o(\log \sigma) + 2d \log d$

Top-k strings in $O(|P| + k \log k)$

Solution based on Compressed Permuterm index

Time $O(|P| \log \log \sigma + occ)$

Space $nH_k + n \cdot o(\log \sigma)$ for $\sigma = \omega(\log^c n)$

Open Problems

Solution based on Patricia tries and Perfect

Time $O(|P| + occ)$

Space $2nH_k + n \cdot o(\log \sigma) + 2d \log d$

Top-k strings in $O(|P| + k \log k)$

Solution based on Compressed Permuterm index

Time $O(|P| \log \log \sigma + occ)$

Space $nH_k + n \cdot o(\log \sigma)$ for $\sigma = \omega(\log^c n)$

Open Problems

Optimal time/space complexities?

Time: $O(|P| \log \sigma / w + occ)$ where w size of a word

Space: $O(n \log \sigma)$

Thank You