Extending the BWT for Cartesian Tree Matching

Eric M. Osterkamp and Dominik Köppl

Motivation – 1

- reversal patterns in stock charts

Inverse Head & Shoulder [Fu+ ‘07]

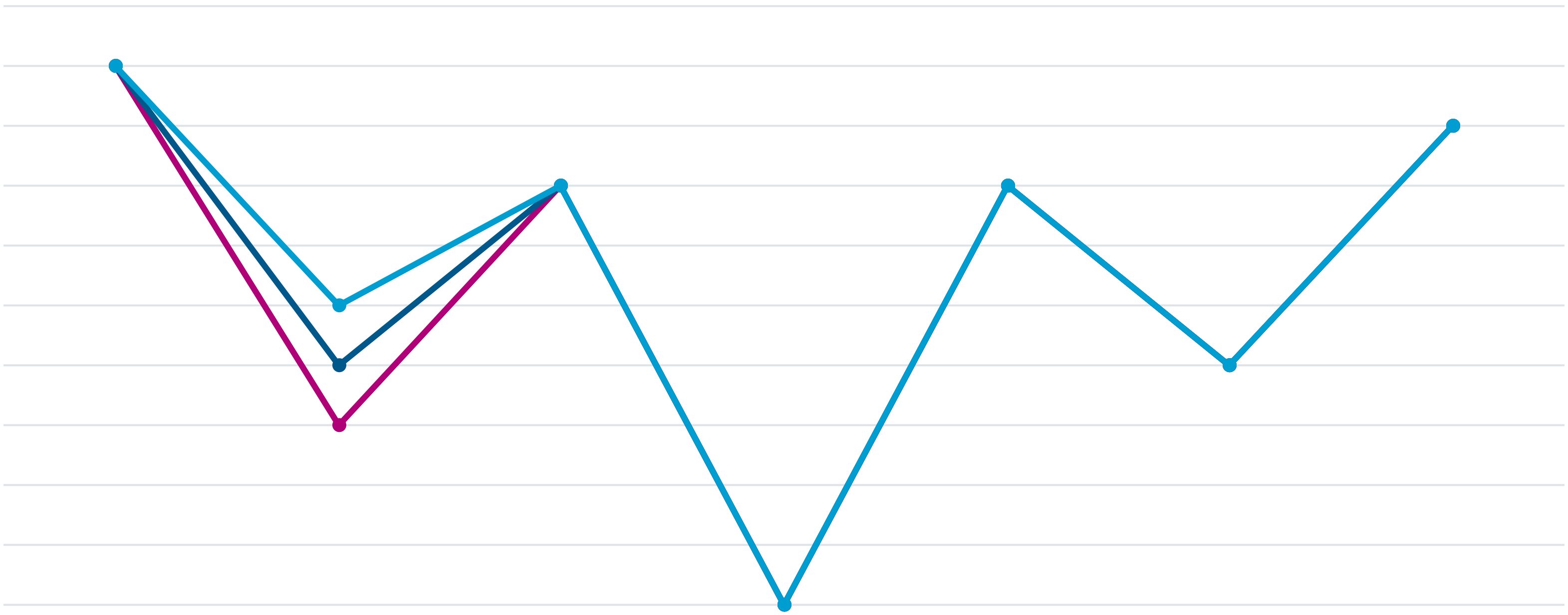
Motivation – 2

- Order-Preserving Matching [Kim+ ‘14] too strict at times

Motivation – 2

- Order-Preserving Matching [Kim+ '14] too strict at times

Inverse Head & Shoulder [Fu+ '07]



Cartesian Tree Matching – 1

- Cartesian Tree Matching introduced by [Park+ ‘20]

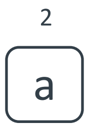
T =

1	2	3	4	5	6
b	a	n	a	n	a

Cartesian Tree Matching – 1

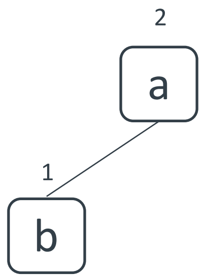
- Cartesian Tree Matching introduced by [Park+ ‘20]

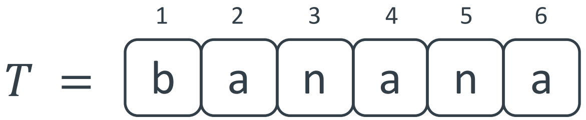
	1	2	3	4	5	6
$T =$	b	a	n	a	n	a



Cartesian Tree Matching – 1

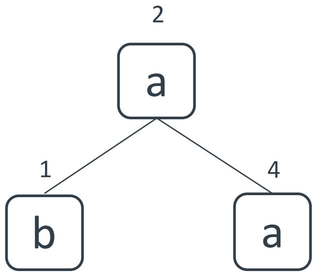
- Cartesian Tree Matching introduced by [Park+ ‘20]





Cartesian Tree Matching – 1

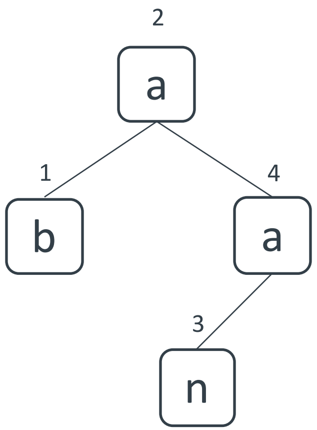
- Cartesian Tree Matching introduced by [Park+ ‘20]



	1	2	3	4	5	6
$T =$	b	a	n	a	n	a

Cartesian Tree Matching – 1

- Cartesian Tree Matching introduced by [Park+ ‘20]

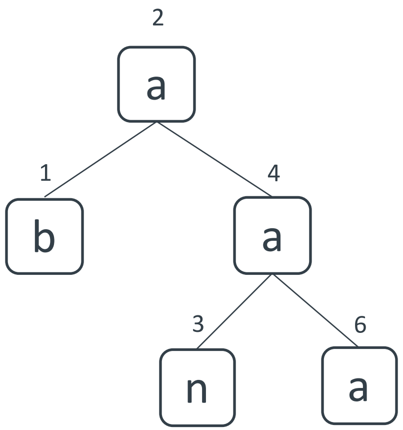


$T =$

1	2	3	4	5	6
b	a	n	a	n	a

Cartesian Tree Matching – 1

- Cartesian Tree Matching introduced by [Park+ ‘20]

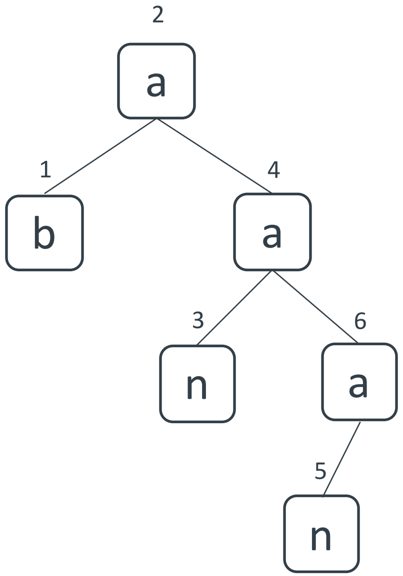


$T =$

1	2	3	4	5	6
b	a	n	a	n	a

Cartesian Tree Matching – 1

- Cartesian Tree Matching introduced by [Park+ ‘20]

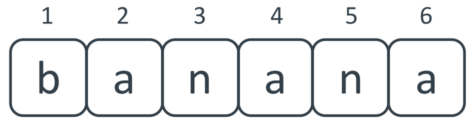


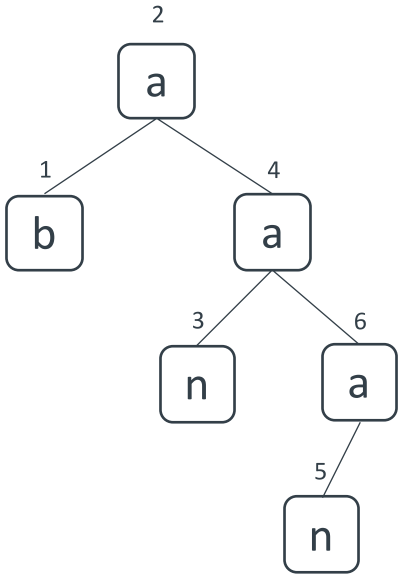
$T =$

1	2	3	4	5	6
b	a	n	a	n	a

Cartesian Tree Matching – 1

- Cartesian Tree Matching introduced by [Park+ ‘20]

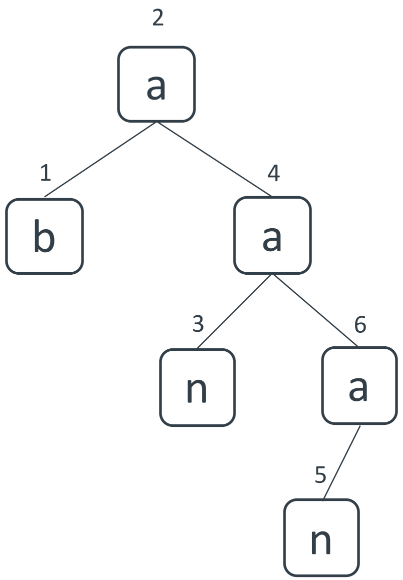




$\mathrm{ct}(T)$

Cartesian Tree Matching – 1

- Cartesian Tree Matching introduced by [Park+ ‘20]



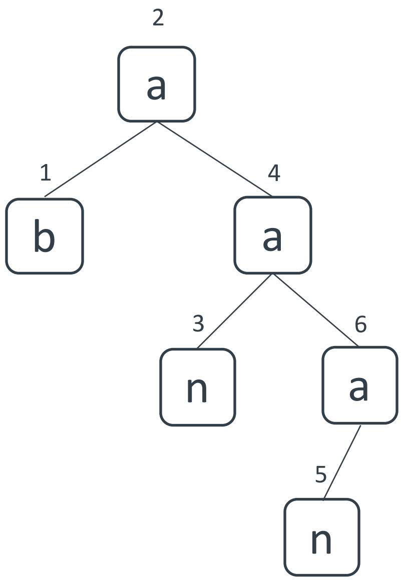
ct(T)

	1	2	3	4	5	6
$T =$	b	a	n	a	n	a

	1	2	3	4	5	6
$S =$	p	a	p	a	y	a

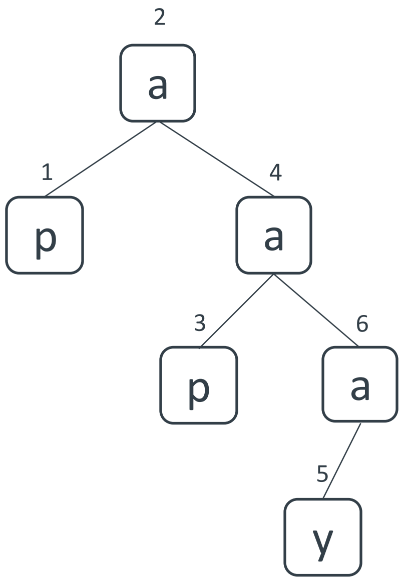
Cartesian Tree Matching – 1

- Cartesian Tree Matching introduced by [Park+ '20]



$\mathrm{ct}(T)$

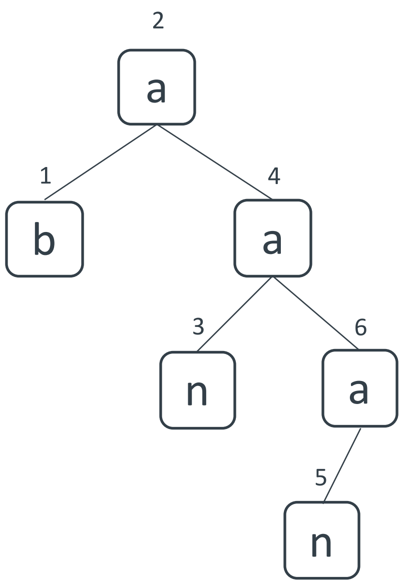
	1	2	3	4	5	6
$T =$	b	a	n	a	n	a
$S =$	p	a	p	a	y	a



$\mathrm{ct}(S)$

Cartesian Tree Matching – 1

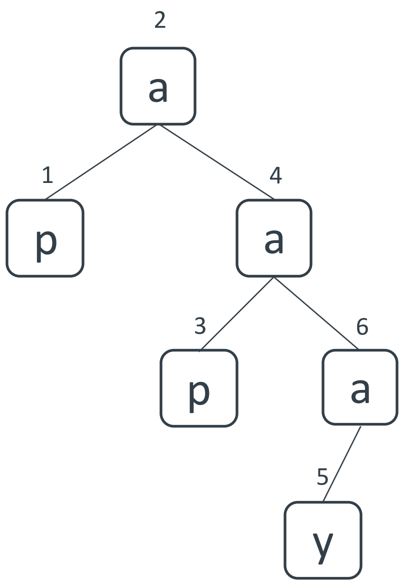
- Cartesian Tree Matching introduced by [Park+ ‘20]



ct(T)

	1	2	3	4	5	6
$T =$	b	a	n	a	n	a
$S =$	p	a	p	a	y	a

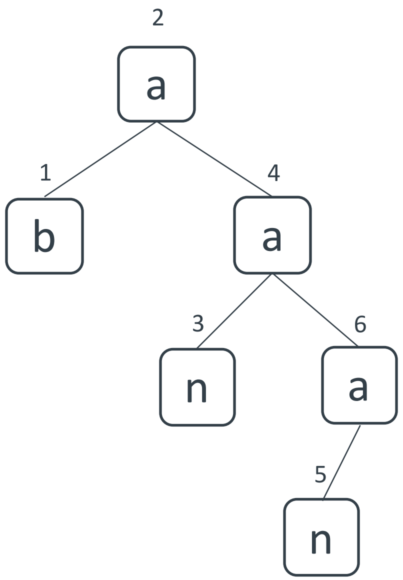
$$T =_{\text{ct}} S$$



ct(S)

Cartesian Tree Matching – 1

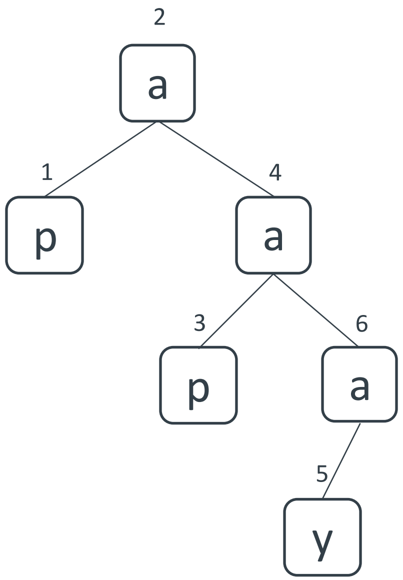
- Cartesian Tree Matching introduced by [Park+ '20]



$\mathrm{ct}(T)$

	1	2	3	4	5	6
$T =$	b	a	n	a	n	a
$S =$	p	a	p	a	y	a
$R =$	l	y	c	h	e	e

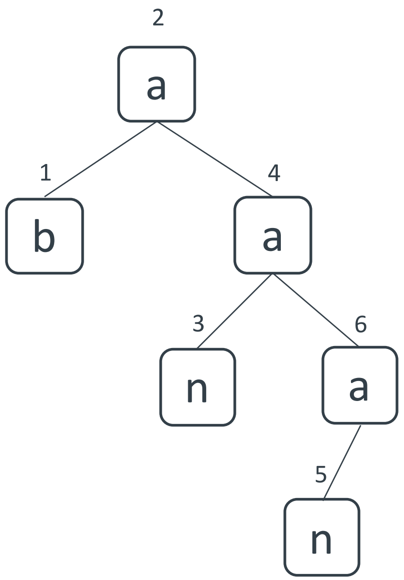
$$T =_{\mathrm{ct}} S$$



$\mathrm{ct}(S)$

Cartesian Tree Matching – 1

- Cartesian Tree Matching introduced by [Park+ ‘20]



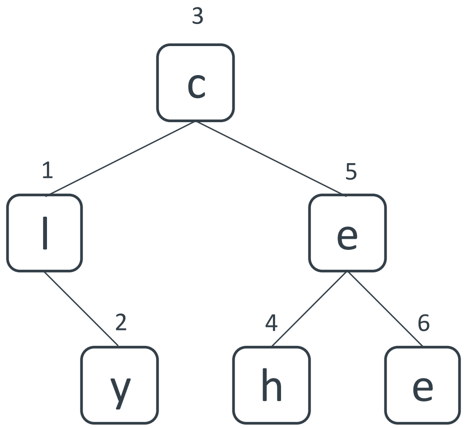
ct(T)

	1	2	3	4	5	6
$T =$	b	a	n	a	n	a

	1	2	3	4	5	6
$S =$	p	a	p	a	y	a

	1	2	3	4	5	6
$R =$	l	y	c	h	e	e

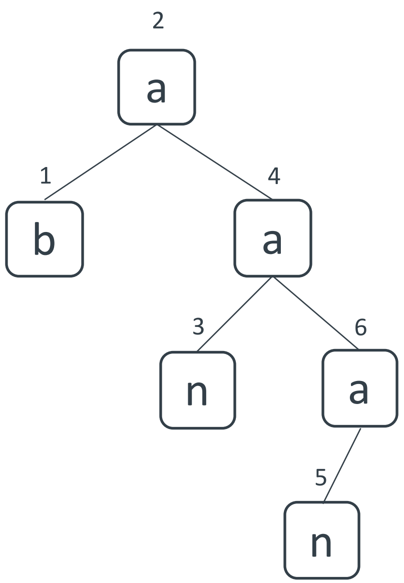
$$T =_{\text{ct}} S$$



ct(R)

Cartesian Tree Matching – 1

- Cartesian Tree Matching introduced by [Park+ ‘20]



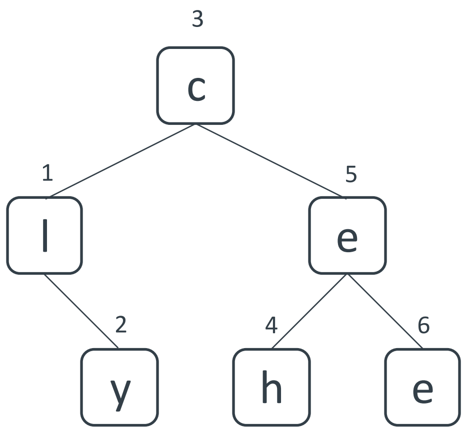
$\mathrm{ct}(T)$

	1	2	3	4	5	6
$T =$	b	a	n	a	n	a

	1	2	3	4	5	6
$S =$	p	a	p	a	y	a

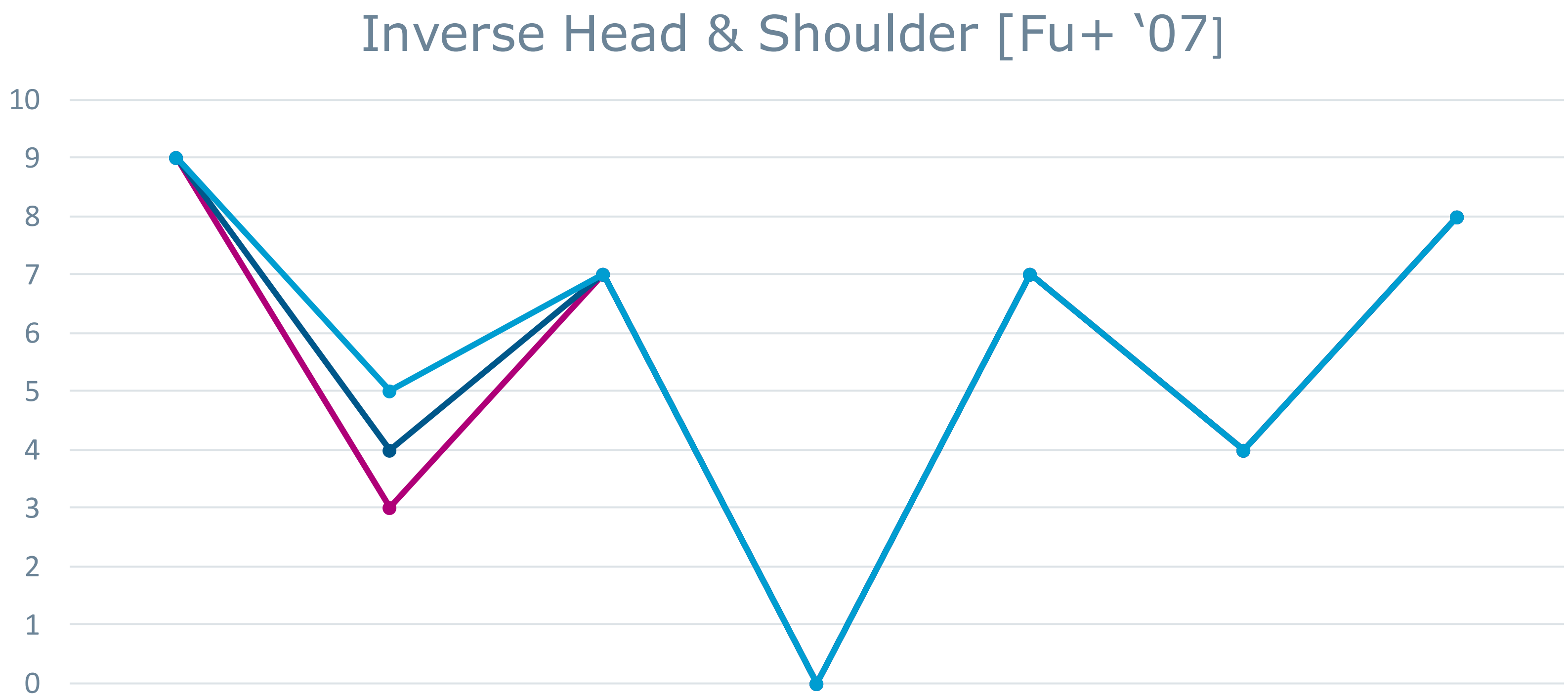
	1	2	3	4	5	6
$R =$	l	y	c	h	e	e

$$T =_{\mathrm{ct}} S \neq_{\mathrm{ct}} R$$

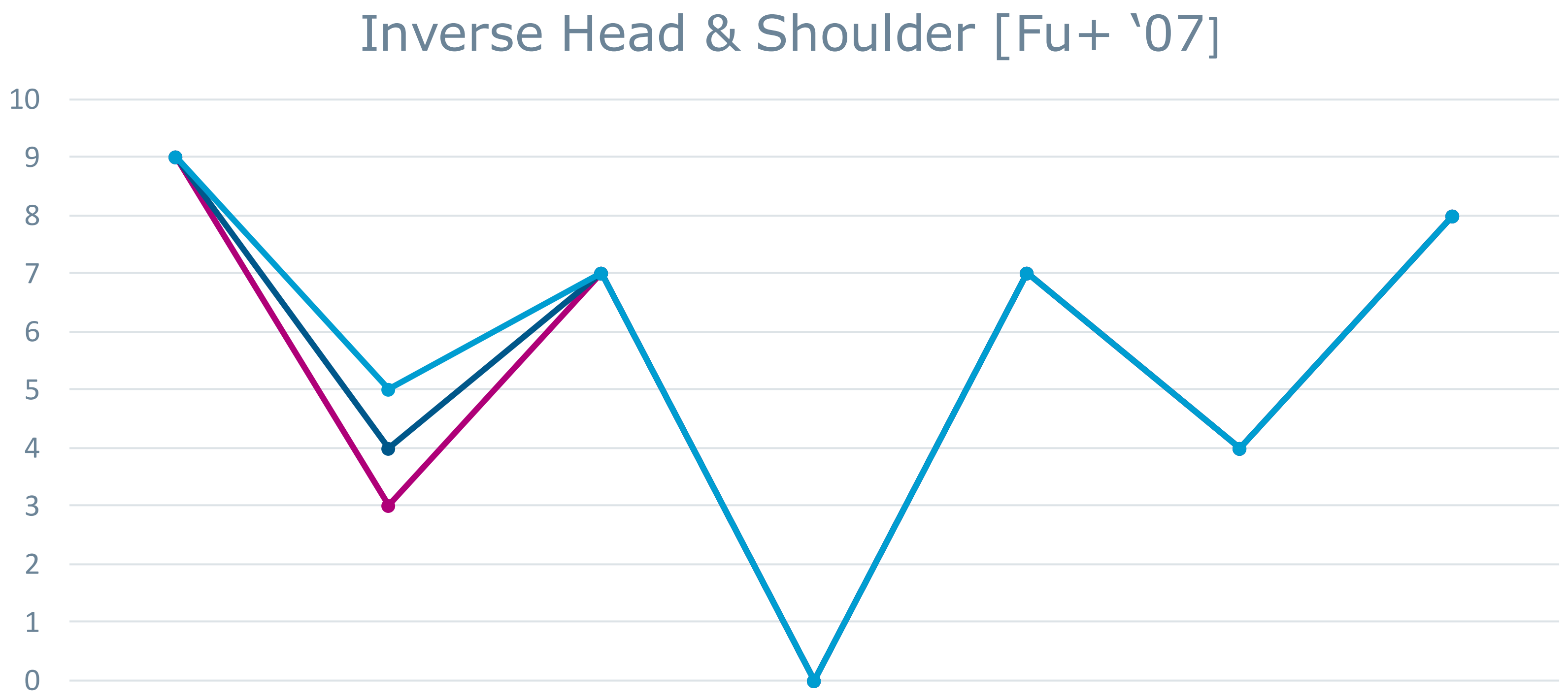


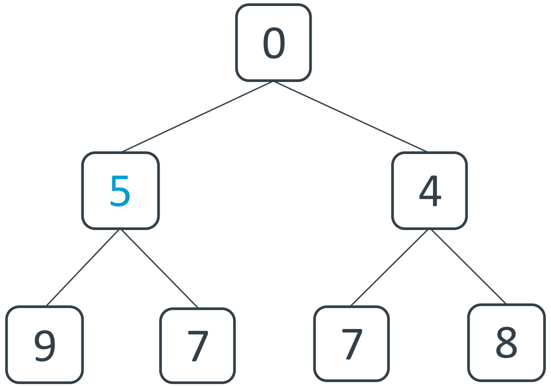
$\mathrm{ct}(R)$

Cartesian Tree Matching – 2



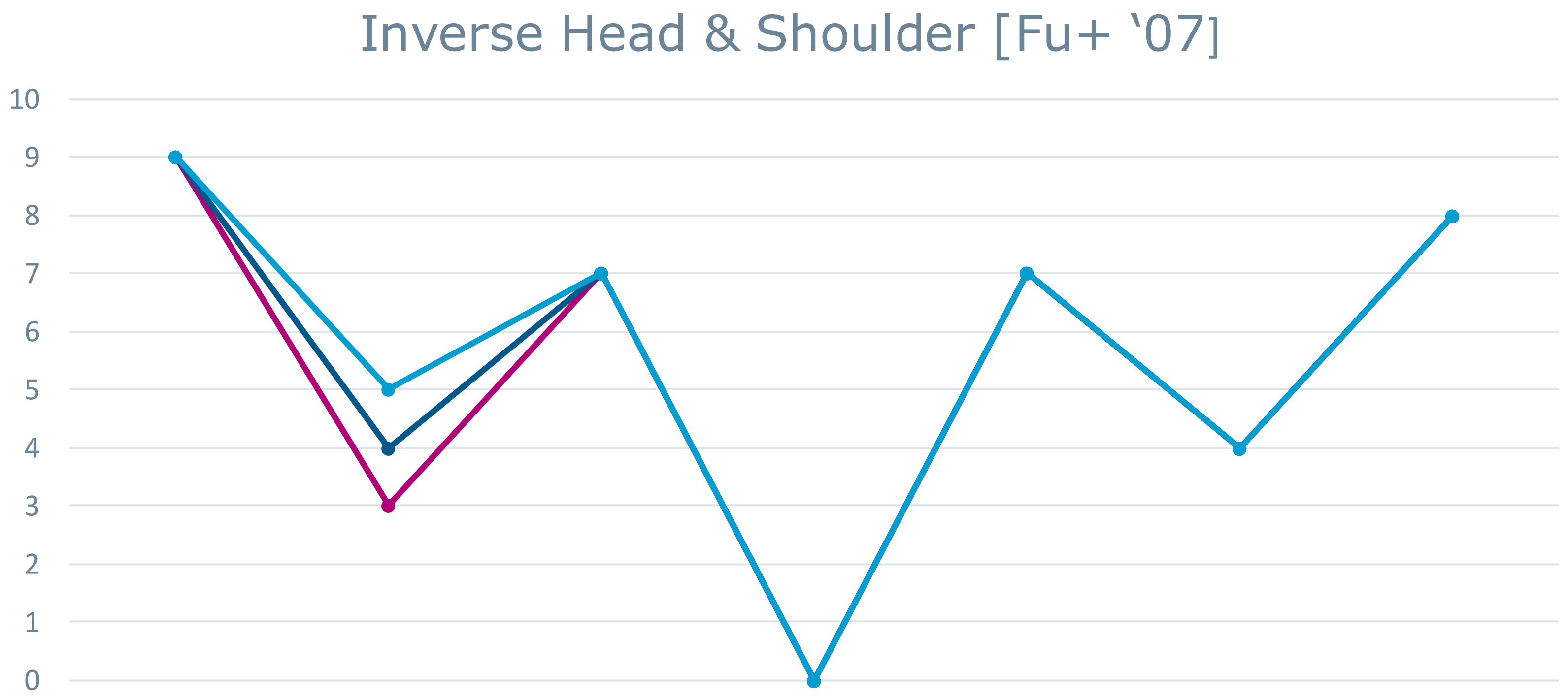
Cartesian Tree Matching – 2

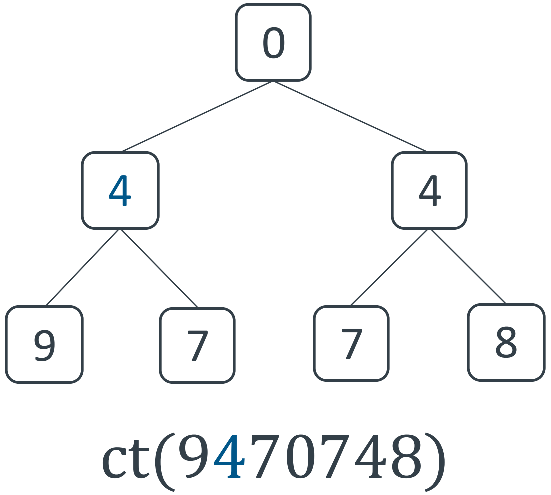




ct(9570748)

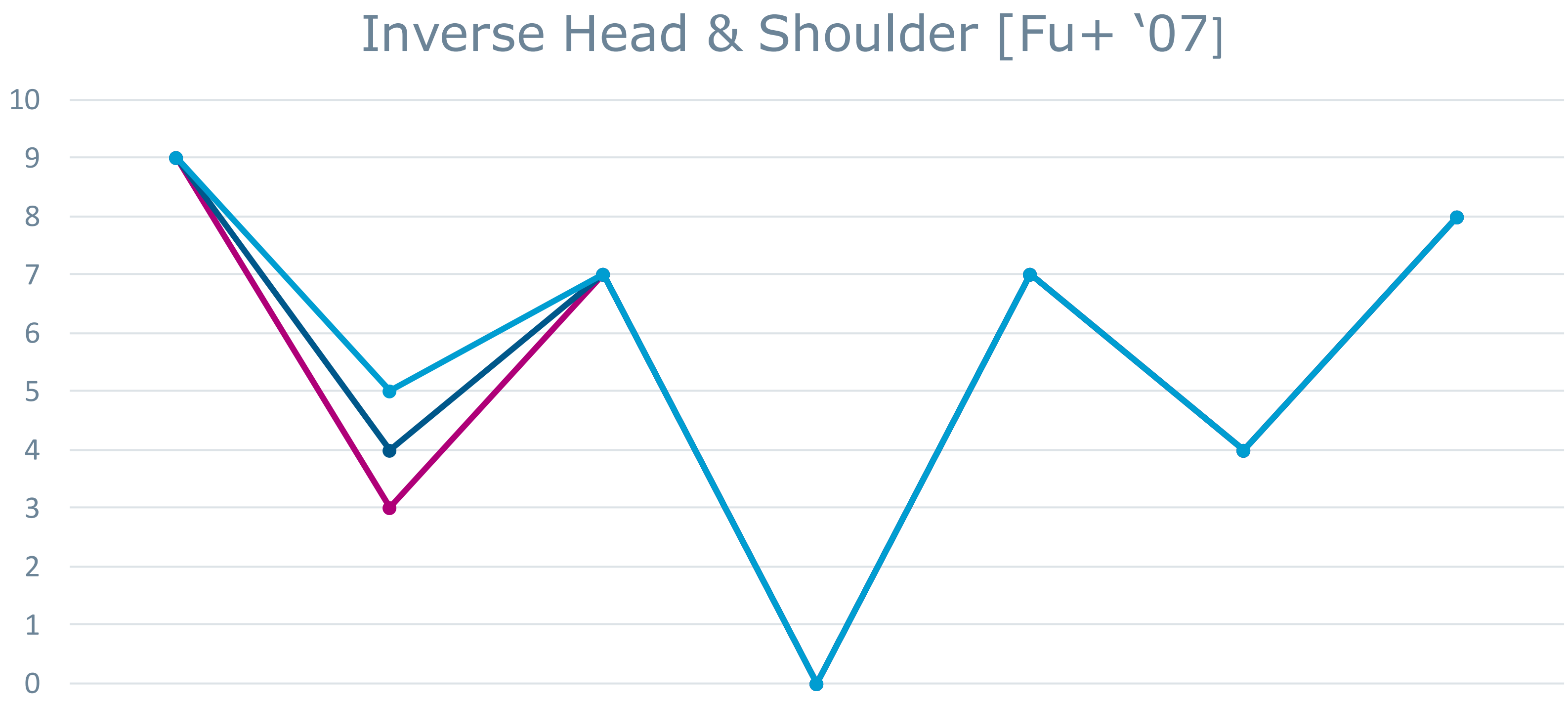
Cartesian Tree Matching – 2

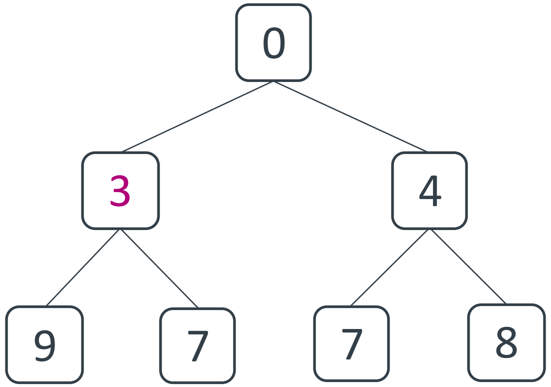




ct(9470748)

Cartesian Tree Matching – 2





$\mathrm{ct}(9370748)$

Cartesian Tree Matching – 3

- pattern P occurs in text T if a substring of T ct-matches P
- CTPM: count all occurrences of P in T

Cartesian Tree Matching – 3

- pattern P occurs in text T if a substring of T ct-matches P
- CTPM: count all occurrences of P in T

$P =$

k	i	w	i

$T =$

1	2	3	4	5	6
b	a	n	a	n	a

Cartesian Tree Matching – 3

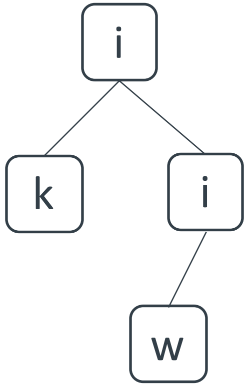
- pattern P occurs in text T if a substring of T ct-matches P
- CTPM: count all occurrences of P in T

$P =$

k	i	w	i

$T =$

1	2	3	4	5	6
b	a	n	a	n	a



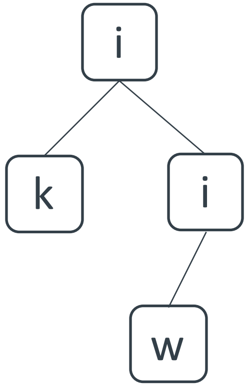
$\mathrm{ct}(P)$

Cartesian Tree Matching – 3

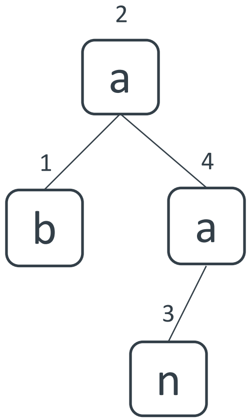
- pattern P occurs in text T if a substring of T ct-matches P
- CTPM: count all occurrences of P in T

P = k i w i

	1	2	3	4	5	6
T =	b	a	n	a	n	a



$\mathrm{ct}(P)$



$\mathrm{ct}(T[1..4])$

Cartesian Tree Matching – 3

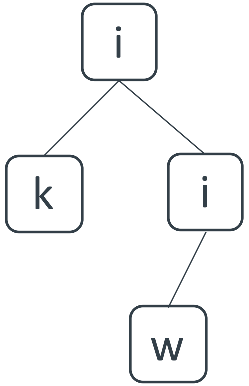
- pattern P occurs in text T if a substring of T ct-matches P
- CTPM: count all occurrences of P in T

$P =$

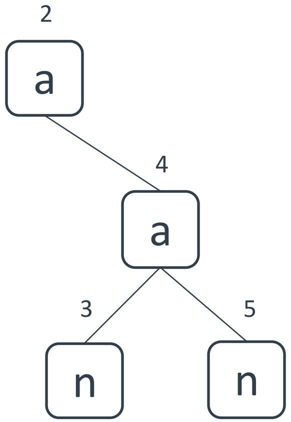
k	i	w	i

$T =$

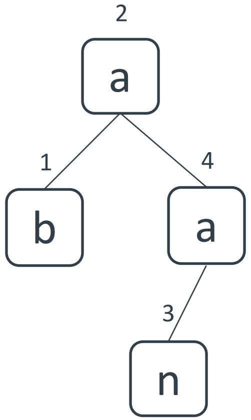
1	2	3	4	5	6
b	a	n	a	n	a



$\mathrm{ct}(P)$



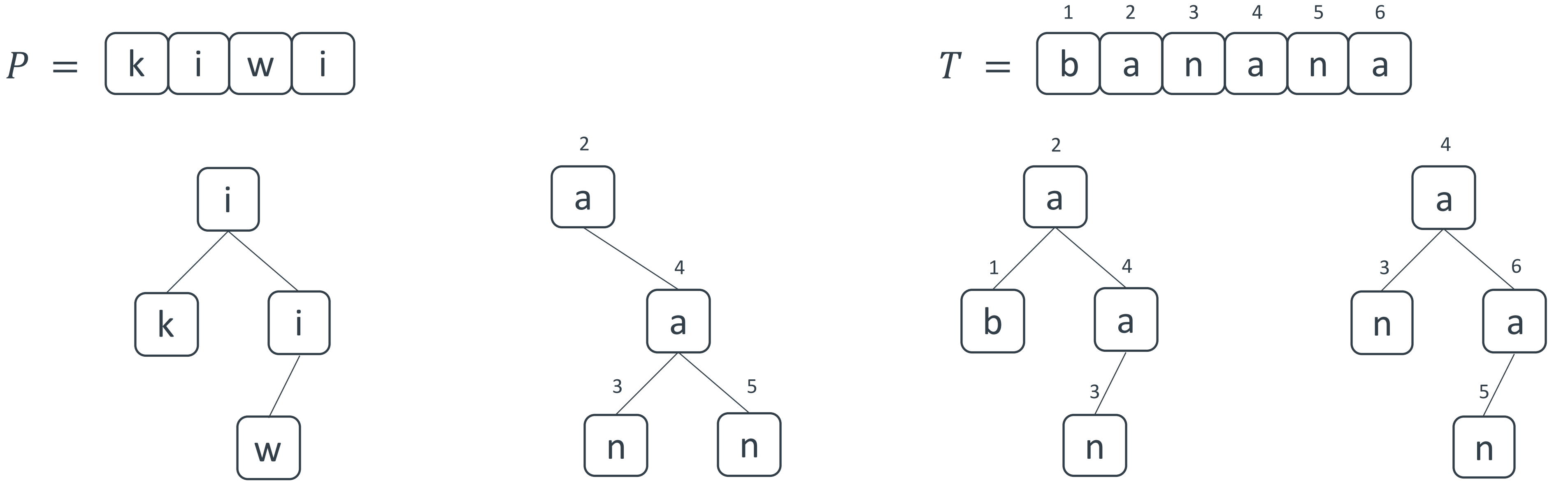
$\mathrm{ct}(T[2..5])$



$\mathrm{ct}(T[1..4])$

Cartesian Tree Matching – 3

- pattern P occurs in text T if a substring of T ct-matches P
- CTPM: count all occurrences of P in T

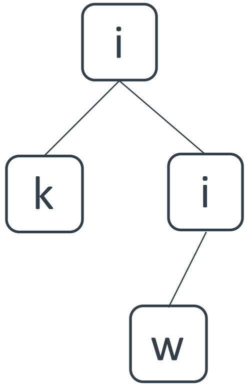


$\mathrm{ct}(P)$ $\mathrm{ct}(T[2..5])$ $\mathrm{ct}(T[1..4])$ $\mathrm{ct}(T[3..6])$

Cartesian Tree Matching – 3

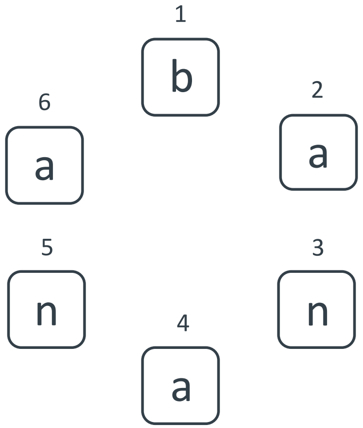
- pattern P occurs in text T if a substring of T ct-matches P
- CTPM: count all occurrences of P in T

P = k i w i



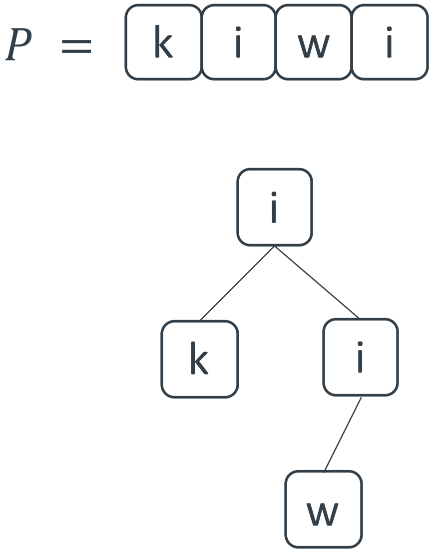
ct(P)

T =

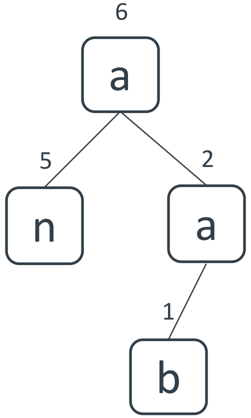


Cartesian Tree Matching – 3

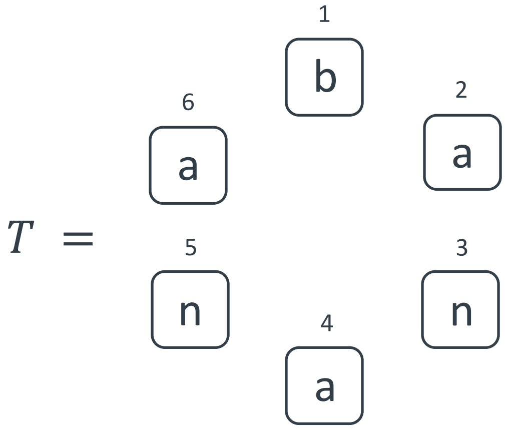
- pattern P occurs in text T if a substring of T ct-matches P
- CTPM: count all occurrences of P in T



$\mathrm{ct}(P)$



$\mathrm{ct}(T[5..6] \cdot T[1..2])$



Cartesian Tree Matching – 4

Data Structure	Bits of Space required	Additional Space for Construction	Construction Time	Time for CTPM	Reference
Suffix Tree	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(m \log \sigma)$	[Park+ ‘20]
Position Heap	$O(n \log n)$	$O(n \log n)$	$O(n \log \sigma)$	$O(m\sigma + m \log m + occ)$	[Nishimoto+ ‘21]
FM-Index (BWT)	$3n + o(n)$	$O(n \log n)$	$O(n \log n)$	$O(m)$	[Kim and Cho ‘21]

- σ alphabet size
- occ total number of occurrences
- n total text length
- m pattern length

Cartesian Tree Matching – 4

Data Structure	Bits of Space required	Additional Space for Construction	Construction Time	Time for CTPM	Reference
Suffix Tree	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(m \log \sigma)$	[Park+ ‘20]
Position Heap	$O(n \log n)$	$O(n \log n)$	$O(n \log \sigma)$	$O(m\sigma + m \log m + occ)$	[Nishimoto+ ‘21]
FM-Index (BWT)	$3n + o(n)$	$O(n \log n)$	$O(n \log n)$	$O(m)$	[Kim and Cho ‘21]
FM-Index (eBWT)	$O(n \log \sigma)$	$O(n \log \sigma)$	$O\left(n \frac{\log \sigma \log n}{\log \log n}\right)$	$O\left(m \frac{\log \sigma \log n}{\log \log n}\right)$	this paper
FM-Index (eBWT)	$3n + o(n)$	$O(n \log \sigma)$	$O\left(n \frac{\log \sigma \log n}{\log \log n}\right)$	$O(m)$	full paper

- σ alphabet size
- occ total number of occurrences
- n total text length
- m pattern length

Parent Distance Encoding

- coined by [Park+ ‘20]

Parent Distance Encoding

- coined by [Park+ ‘20]

$T =$

1	2	3	4	5	6
b	a	n	a	n	a

Parent Distance Encoding

- coined by [Park+ ‘20]
- ∞ is a special symbol larger than any integer

$T =$

1	2	3	4	5	6
b	a	n	a	n	a

$\langle T \rangle =$

1
∞

Parent Distance Encoding

- coined by [Park+ ‘20]
- ∞ is a special symbol larger than any integer

$T =$

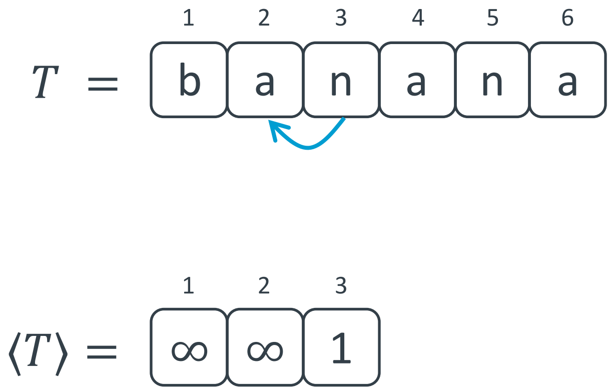
1	2	3	4	5	6
b	a	n	a	n	a

$\langle T \rangle =$

1	2
∞	∞

Parent Distance Encoding

- coined by [Park+ ‘20]
- ∞ is a special symbol larger than any integer



$T =$

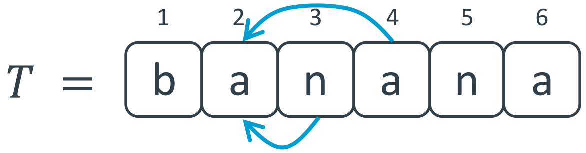
1	2	3	4	5	6
b	a	n	a	n	a

$\langle T \rangle =$

1	2	3
∞	∞	1

Parent Distance Encoding

- coined by [Park+ ‘20]
- ∞ is a special symbol larger than any integer

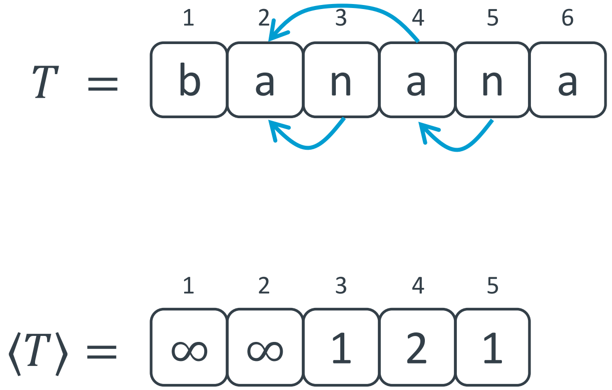


$\langle T \rangle =$

1	2	3	4
∞	∞	1	2

Parent Distance Encoding

- coined by [Park+ ‘20]
- ∞ is a special symbol larger than any integer

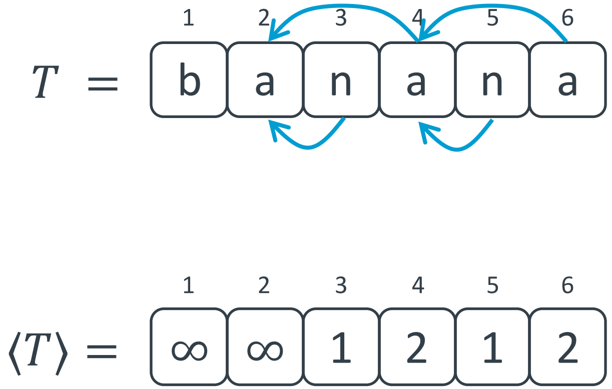


	1	2	3	4	5	6
T =	b	a	n	a	n	a

	1	2	3	4	5
$\langle T \rangle$ =	∞	∞	1	2	1

Parent Distance Encoding

- coined by [Park+ ‘20]
- ∞ is a special symbol larger than any integer



$T =$

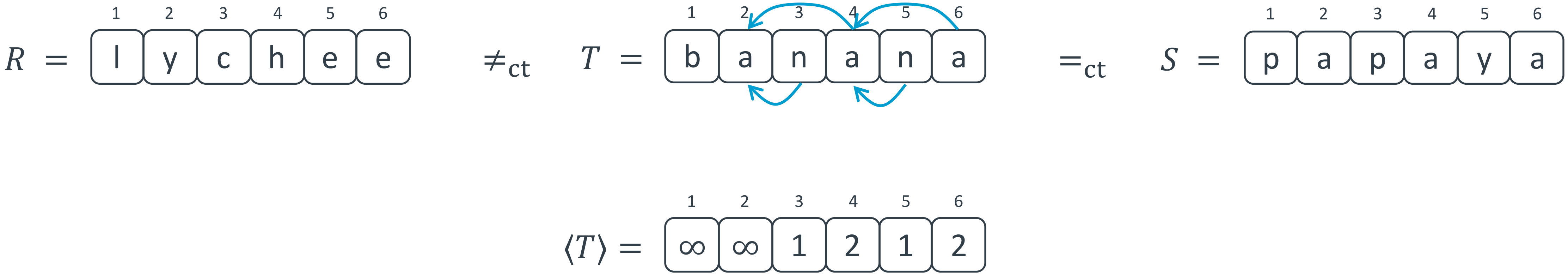
1	2	3	4	5	6
b	a	n	a	n	a

$\langle T \rangle =$

1	2	3	4	5	6
∞	∞	1	2	1	2

Parent Distance Encoding

- coined by [Park+ ‘20]
- ∞ is a special symbol larger than any integer



$R =$

1	2	3	4	5	6
l	y	c	h	e	e

$\neq_{\text{ct}}$ $T =$

1	2	3	4	5	6
b	a	n	a	n	a

$=_{\text{ct}}$ $S =$

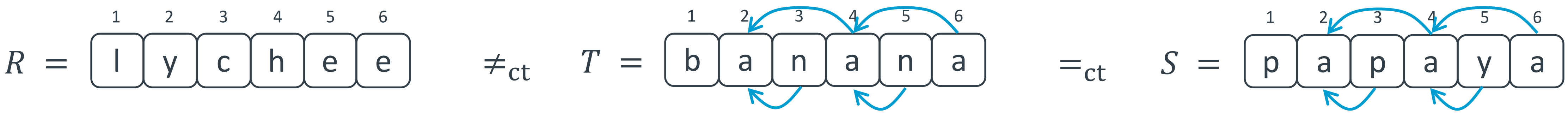
1	2	3	4	5	6
p	a	p	a	y	a

$\langle T \rangle =$

1	2	3	4	5	6
∞	∞	1	2	1	2

Parent Distance Encoding

- coined by [Park+ ‘20]
- ∞ is a special symbol larger than any integer



	1	2	3	4	5	6
$R =$	l	y	c	h	e	e

$\neq_{\mathrm{ct}}$

	1	2	3	4	5	6
$T =$	b	a	n	a	n	a

$=_{\mathrm{ct}}$

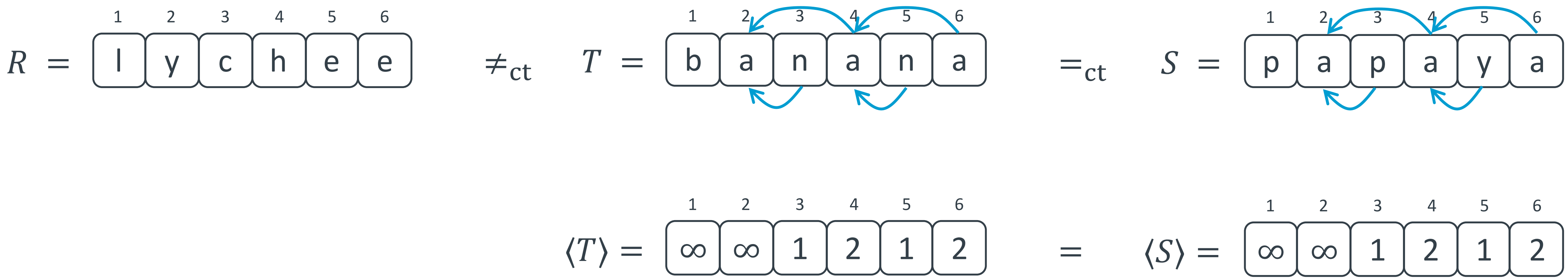
	1	2	3	4	5	6
$S =$	p	a	p	a	y	a

	1	2	3	4	5	6
$\langle T \rangle =$	∞	∞	1	2	1	2

	1	2	3	4	5	6
$\langle S \rangle =$	∞	∞	1	2	1	2

Parent Distance Encoding

- coined by [Park+ ‘20]
- ∞ is a special symbol larger than any integer

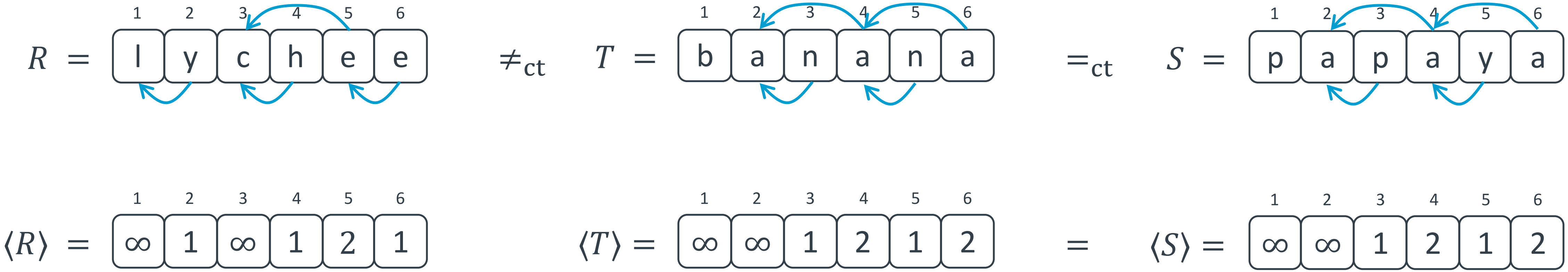


$$R = \begin{array}{|c|c|c|c|c|c|} \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4} & \multicolumn{1}{c}{5} & \multicolumn{1}{c}{6} \\ \hline \text{l} & \text{y} & \text{c} & \text{h} & \text{e} & \text{e} \\ \hline \end{array} \neq_{\text{ct}} T = \begin{array}{|c|c|c|c|c|c|} \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4} & \multicolumn{1}{c}{5} & \multicolumn{1}{c}{6} \\ \hline \text{b} & \text{a} & \text{n} & \text{a} & \text{n} & \text{a} \\ \hline \end{array} =_{\text{ct}} S = \begin{array}{|c|c|c|c|c|c|} \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4} & \multicolumn{1}{c}{5} & \multicolumn{1}{c}{6} \\ \hline \text{p} & \text{a} & \text{p} & \text{a} & \text{y} & \text{a} \\ \hline \end{array}$$

$$\langle T \rangle = \begin{array}{|c|c|c|c|c|c|} \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4} & \multicolumn{1}{c}{5} & \multicolumn{1}{c}{6} \\ \hline \infty & \infty & 1 & 2 & 1 & 2 \\ \hline \end{array} = \langle S \rangle = \begin{array}{|c|c|c|c|c|c|} \multicolumn{1}{c}{1} & \multicolumn{1}{c}{2} & \multicolumn{1}{c}{3} & \multicolumn{1}{c}{4} & \multicolumn{1}{c}{5} & \multicolumn{1}{c}{6} \\ \hline \infty & \infty & 1 & 2 & 1 & 2 \\ \hline \end{array}$$

Parent Distance Encoding

- coined by [Park+ ‘20]
- ∞ is a special symbol larger than any integer

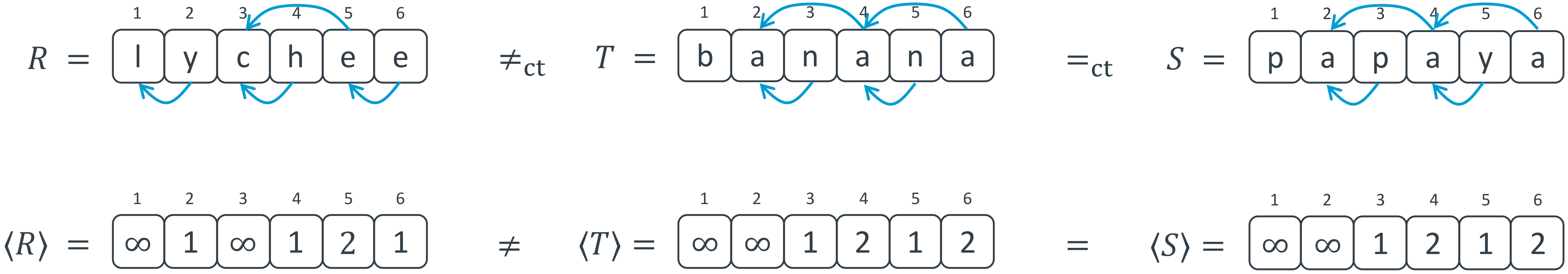


$R =$ l y c h e e $\neq_{\mathrm{ct}}$ $T =$ b a n a n a $=_{\mathrm{ct}}$ $S =$ p a p a y a

$\langle R \rangle = \infty\ 1\ \infty\ 1\ 2\ 1$ $\quad \langle T \rangle = \infty\ \infty\ 1\ 2\ 1\ 2$ $=$ $\langle S \rangle = \infty\ \infty\ 1\ 2\ 1\ 2$

Parent Distance Encoding

- coined by [Park+ '20]
- ∞ is a special symbol larger than any integer



$R = \boxed{\text{l}}\boxed{\text{y}}\boxed{\text{c}}\boxed{\text{h}}\boxed{\text{e}}\boxed{\text{e}} \neq_{\text{ct}} T = \boxed{\text{b}}\boxed{\text{a}}\boxed{\text{n}}\boxed{\text{a}}\boxed{\text{n}}\boxed{\text{a}} =_{\text{ct}} S = \boxed{\text{p}}\boxed{\text{a}}\boxed{\text{p}}\boxed{\text{a}}\boxed{\text{y}}\boxed{\text{a}}$

$\langle R \rangle = \boxed{\infty}\boxed{1}\boxed{\infty}\boxed{1}\boxed{2}\boxed{1} \neq \langle T \rangle = \boxed{\infty}\boxed{\infty}\boxed{1}\boxed{2}\boxed{1}\boxed{2} = \langle S \rangle = \boxed{\infty}\boxed{\infty}\boxed{1}\boxed{2}\boxed{1}\boxed{2}$

Problem Statement

- the conjugates of a string are its rotations
- identify conjugates of $T_1, \ldots, T_d$ by their starting position in $T = T_1 \cdot \ldots \cdot T_d$

Problem Statement

- the conjugates of a string are its rotations
- identify conjugates of $T_1, \dots, T_d$ by their starting position in $T = T_1 \cdot \dots \cdot T_d$

	1	2	3		4	5	6	7		8	9	10	11	12
$T =$	f	i	g	$\cdot$	k	i	w	i	$\cdot$	a	p	p	l	e

Problem Statement

- the conjugates of a string are its rotations
- identify conjugates of $T_1, \ldots, T_d$ by their starting position in $T = T_1 \cdot \ldots \cdot T_d$

	1	2	3		4	5	6	7		8	9	10	11	12	
$T =$	f	i	g	$\cdot$	k	i	w	i	$\cdot$	a	p	p	l	e	$\Rightarrow \mathrm{C}(6) = \mathrm{wiki}$

Problem Statement

- the conjugates of a string are its rotations
- identify conjugates of $T_1, \dots, T_d$ by their starting position in $T = T_1 \cdot \ldots \cdot T_d$

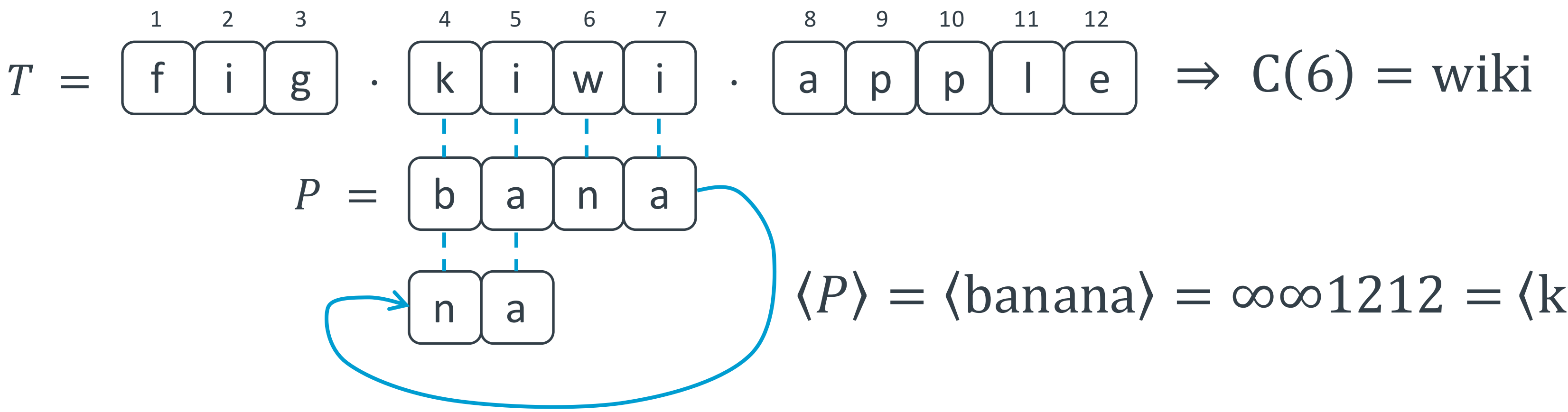
	1	2	3		4	5	6	7		8	9	10	11	12
$T =$	f	i	g	$\cdot$	k	i	w	i	$\cdot$	a	p	p	l	e

$\Rightarrow \mathrm{C}(6) = \mathrm{wiki}$

- let $X^\omega = X \cdot X \cdot X \cdot \ldots$ and P a pattern
- circular CTPM: count the conjugates of $T_1, \dots, T_d$ such that $\mathrm{C}(i)^\omega[..|P|] =_{\mathrm{ct}} P$

Problem Statement

- the conjugates of a string are its rotations
- identify conjugates of $T_1, \ldots, T_d$ by their starting position in $T = T_1 \cdot \ldots \cdot T_d$



$\langle P \rangle = \langle \mathrm{banana} \rangle = \infty\infty 1212 = \langle \mathrm{kiwiki} \rangle = \langle \mathrm{C}(4)^{\omega}[..6] \rangle$

- let $X^{\omega} = X \cdot X \cdot X \cdot \ldots$ and P a pattern
- circular CTPM: count the conjugates of $T_1, \ldots, T_d$ such that $\mathrm{C}(i)^{\omega}[..|P|] =_{\mathrm{ct}} P$

Sorting Conjugates – 1

- sort conjugates such that occurrences of a pattern are consecutive
- sort encoded conjugates lexicographically like [Kim and Cho ‘21]?

	1	2	3		4	5	6	7		8	9	10	11	12
$T =$	f	i	g	·	k	i	w	i	·	a	p	p	l	e

i	$\mathrm{C}(i)$	$\langle \mathrm{C}(i) \rangle$
8	apple	∞1134
1	fig	∞12
5	iwik	∞121
7	ikiw	∞121
9	pplea	∞1∞∞∞
3	gfi	∞∞1
12	eappl	∞∞113
4	kiwi	∞∞12
6	wiki	∞∞12
2	igf	∞∞∞
11	leapp	∞∞∞11
10	pleap	∞∞∞∞1

break ties by index

Sorting Conjugates – 1

- sort conjugates such that occurrences of a pattern are consecutive
- sort encoded conjugates lexicographically like [Kim and Cho ‘21]?

$T =$

1	2	3		4	5	6	7		8	9	10	11	12
f	i	g	·	k	i	w	i	·	a	p	p	l	e

$P =$

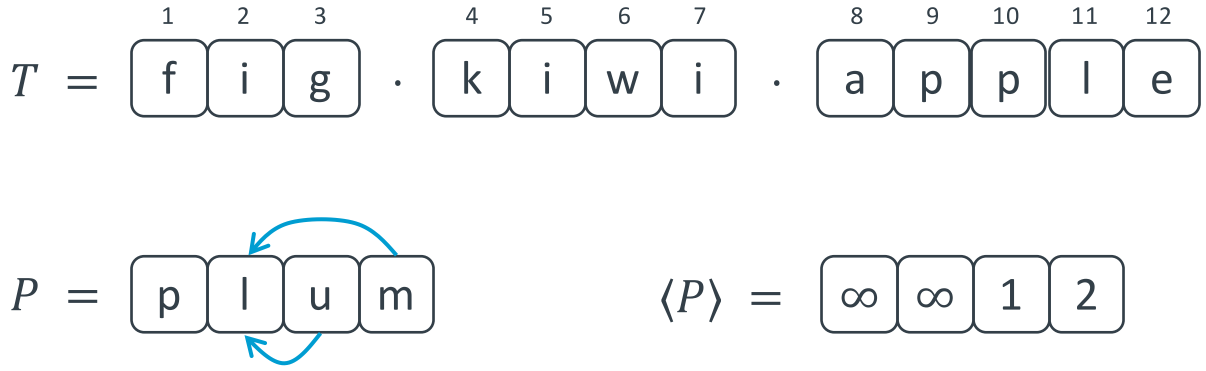
p	l	u	m

break ties by index

i	$\mathsf{C}(i)$	$\langle\mathsf{C}(i)\rangle$
8	apple	∞1134
1	fig	∞12
5	iwik	∞121
7	ikiw	∞121
9	pplea	∞1∞∞∞
3	gfi	∞∞1
12	eappl	∞∞113
4	kiwi	∞∞12
6	wiki	∞∞12
2	igf	∞∞∞
11	leapp	∞∞∞11
10	pleap	∞∞∞∞1

Sorting Conjugates – 1

- sort conjugates such that occurrences of a pattern are consecutive
- sort encoded conjugates lexicographically like [Kim and Cho ‘21]?

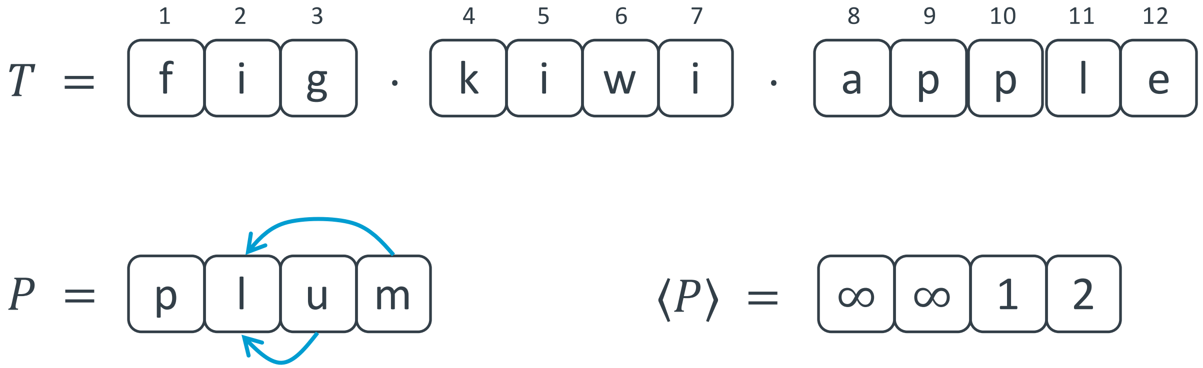


break ties by index

i	$\mathbf{C}(i)$	$\langle \mathbf{C}(i) \rangle$
8	apple	∞1134
1	fig	∞12
5	iwik	∞121
7	ikiw	∞121
9	pplea	∞1∞∞∞
3	gfi	∞∞1
12	eappl	∞∞113
4	kiwi	∞∞12
6	wiki	∞∞12
2	igf	∞∞∞
11	leapp	∞∞∞11
10	pleap	∞∞∞∞1

Sorting Conjugates – 1

- sort conjugates such that occurrences of a pattern are consecutive
- sort encoded conjugates lexicographically like [Kim and Cho '21]?



i	C(i)	⟨C(i)⟩
8	apple	∞1134
1	fig	∞12
5	iwik	∞121
7	ikiw	∞121
9	pplea	∞1∞∞∞∞
3	gfig	∞∞12
12	eappl	∞∞113
4	kiwi	∞∞12
6	wiki	∞∞12
2	igf	∞∞∞
11	leapp	∞∞∞11
10	pleap	∞∞∞∞1

break ties by index

Sorting Conjugates – 2

- adapt approach by [Mantaci+ ‘07] for eBWT
- define the rotational parent distance encoding $\langle T \rangle_r$ of a string T:

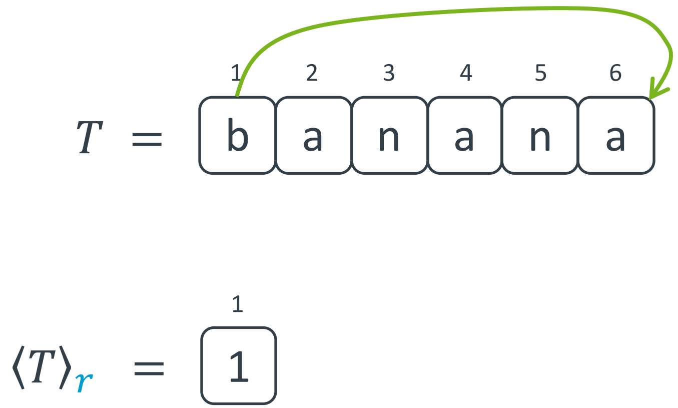
Sorting Conjugates – 2

- adapt approach by [Mantaci+ ‘07] for eBWT
- define the rotational parent distance encoding $\langle T \rangle_r$ of a string T:

	1	2	3	4	5	6
T =	b	a	n	a	n	a

Sorting Conjugates – 2

- adapt approach by [Mantaci+ ‘07] for eBWT
- define the rotational parent distance encoding $\langle T \rangle_r$ of a string T:

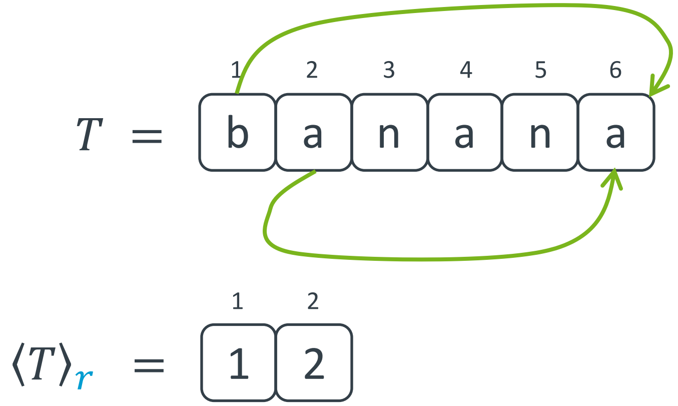


	1	2	3	4	5	6
$T =$	b	a	n	a	n	a

	1
$\langle T \rangle_r =$	1

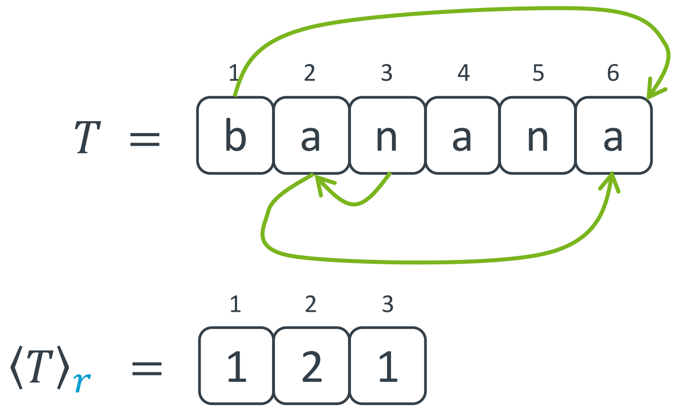
Sorting Conjugates – 2

- adapt approach by [Mantaci+ '07] for eBWT
- define the rotational parent distance encoding $\langle T \rangle_r$ of a string T:



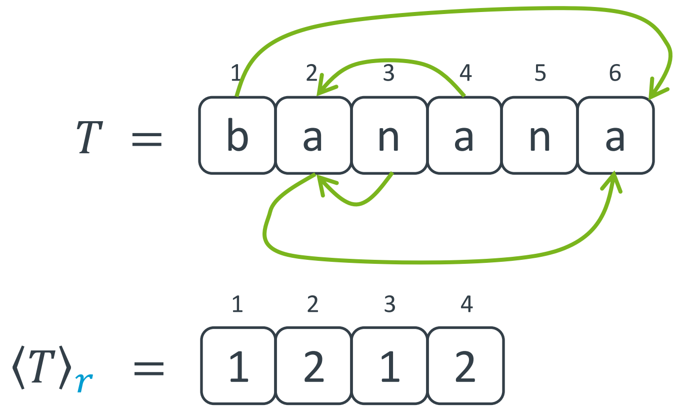
Sorting Conjugates – 2

- adapt approach by [Mantaci+ ‘07] for eBWT
- define the rotational parent distance encoding $\langle T \rangle_r$ of a string T:



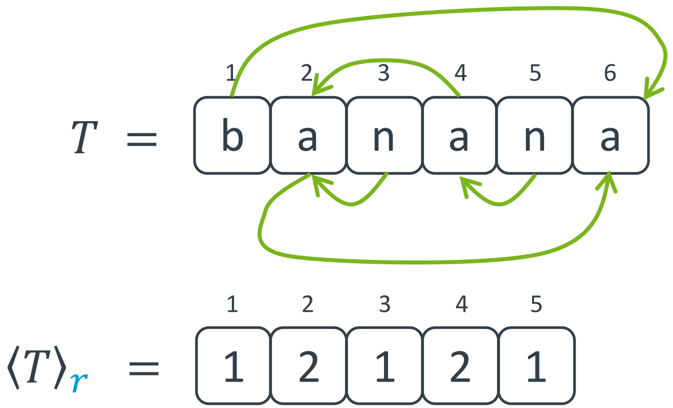
Sorting Conjugates – 2

- adapt approach by [Mantaci+ ‘07] for eBWT
- define the rotational parent distance encoding $\langle T\rangle_r$ of a string T:



Sorting Conjugates – 2

- adapt approach by [Mantaci+ ‘07] for eBWT
- define the rotational parent distance encoding $\langle T \rangle_r$ of a string T:



$T =$

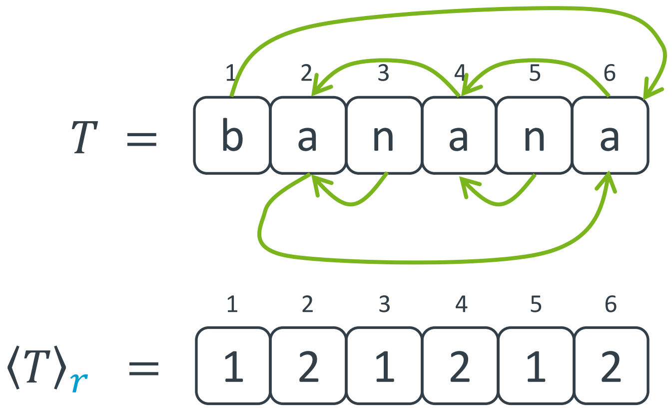
1	2	3	4	5	6
b	a	n	a	n	a

$\langle T \rangle_r =$

1	2	3	4	5
1	2	1	2	1

Sorting Conjugates – 2

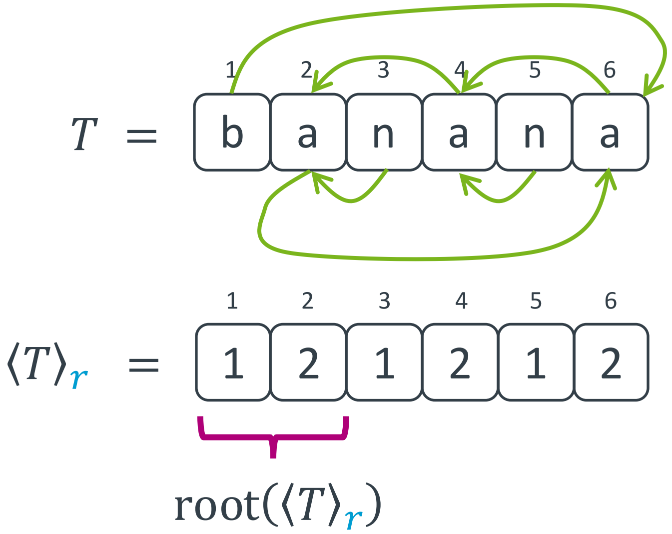
- adapt approach by [Mantaci+ ‘07] for eBWT
- define the rotational parent distance encoding $\langle T\rangle_r$ of a string T:



	1	2	3	4	5	6
T =	b	a	n	a	n	a
$\langle T\rangle_r$ =	1	2	1	2	1	2

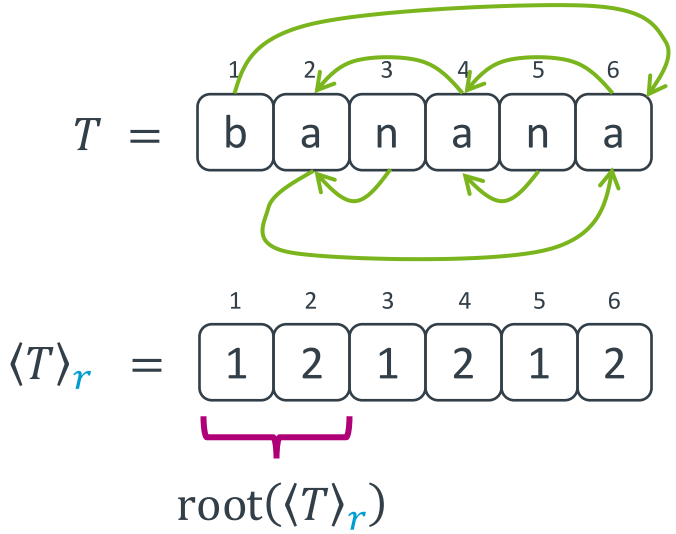
Sorting Conjugates – 2

- adapt approach by [Mantaci+ ‘07] for eBWT
- define the rotational parent distance encoding $\langle T \rangle_r$ of a string T:



Sorting Conjugates – 2

- adapt approach by [Mantaci+ ‘07] for eBWT
- define the rotational parent distance encoding $\langle T\rangle_r$ of a string T:



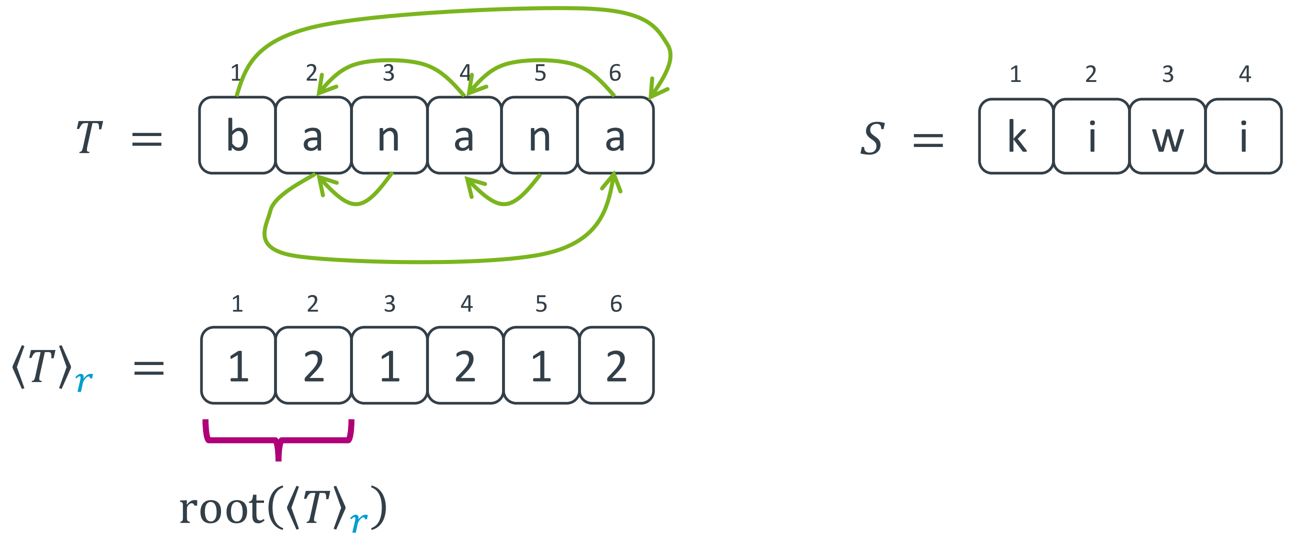
$T =$ b a n a n a

$$\langle T\rangle_r = \underbrace{1\;2}_{\mathrm{root}(\langle T\rangle_r)}\;1\;2\;1\;2$$

- ω-preorder: $X \preccurlyeq_{\omega} Y \Leftrightarrow \mathrm{root}(\langle X\rangle_r) = \mathrm{root}(\langle Y\rangle_r) \vee \exists i{:}\, \langle X^{\omega}[..i]\rangle < \langle Y^{\omega}[..i]\rangle$

Sorting Conjugates – 2

- adapt approach by [Mantaci+ ‘07] for eBWT
- define the rotational parent distance encoding $\langle T\rangle_r$ of a string T:



T = b a n a n a

S = k i w i

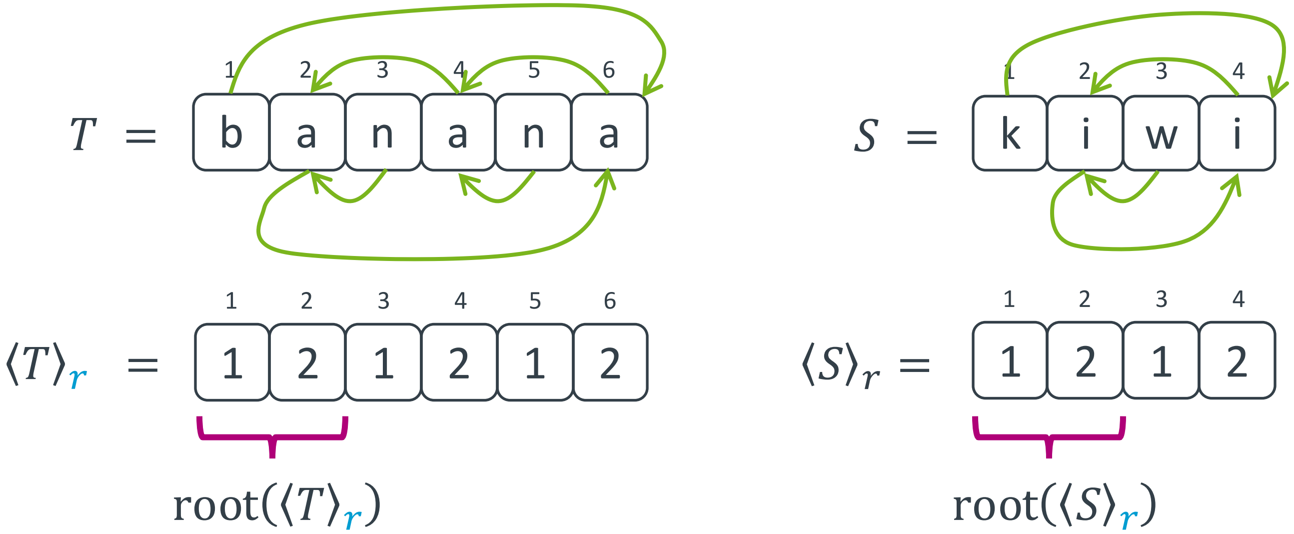
$\langle T\rangle_r$ = 1 2 1 2 1 2

$\mathrm{root}(\langle T\rangle_r)$

- ω-preorder: $X \preccurlyeq_\omega Y \Leftrightarrow \mathrm{root}(\langle X\rangle_r) = \mathrm{root}(\langle Y\rangle_r) \ \vee\ \exists i\colon \langle X^\omega[..i]\rangle < \langle Y^\omega[..i]\rangle$

Sorting Conjugates – 2

- adapt approach by [Mantaci+ '07] for eBWT
- define the rotational parent distance encoding $\langle T\rangle_r$ of a string T:



$T =$ b a n a n a

$S =$ k i w i

$\langle T\rangle_r =$ 1 2 1 2 1 2

$\langle S\rangle_r =$ 1 2 1 2

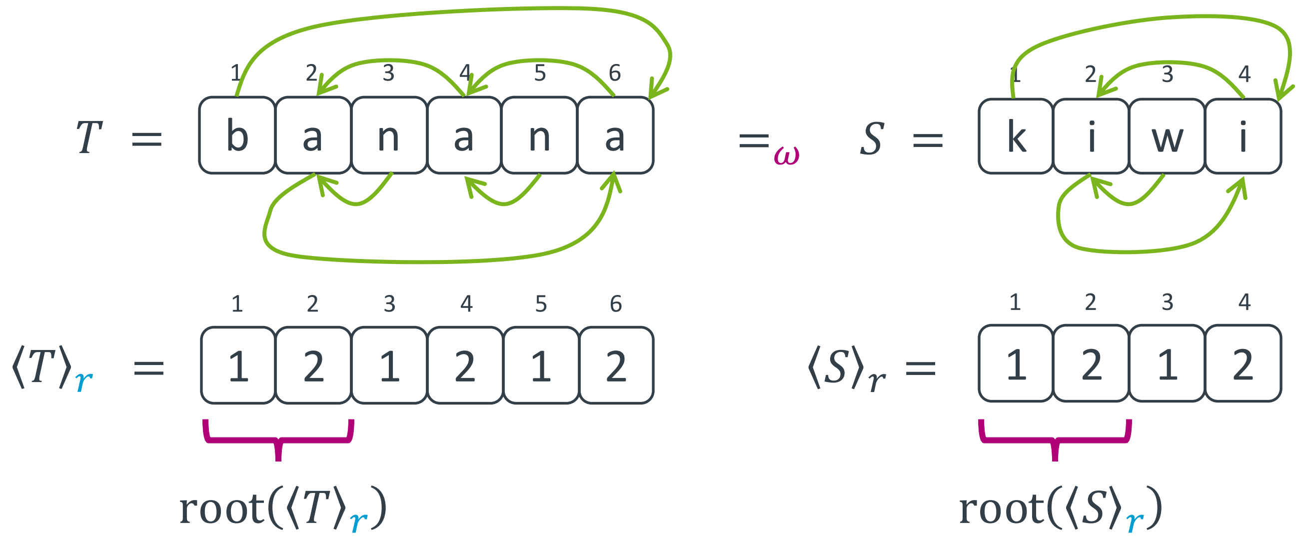
$\mathrm{root}(\langle T\rangle_r)$

$\mathrm{root}(\langle S\rangle_r)$

- ω-preorder: $X \preccurlyeq_\omega Y \Leftrightarrow \mathrm{root}(\langle X\rangle_r) = \mathrm{root}(\langle Y\rangle_r) \lor \exists i{:} \langle X^\omega[..i]\rangle < \langle Y^\omega[..i]\rangle$

Sorting Conjugates – 2

- adapt approach by [Mantaci+ ‘07] for eBWT
- define the rotational parent distance encoding $\langle T\rangle_r$ of a string T:



$T =$ b a n a n a $=_\omega$ $S =$ k i w i

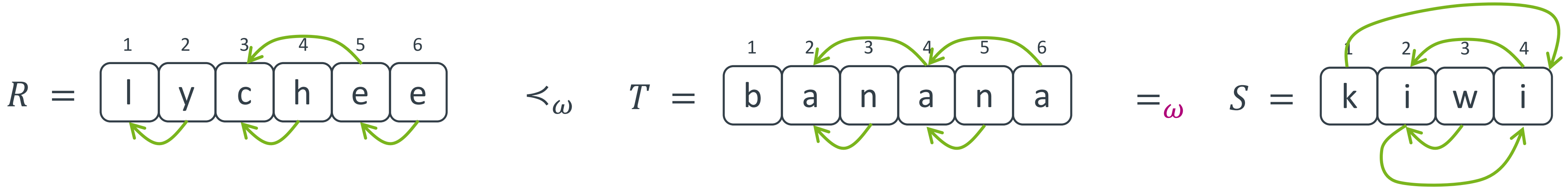
$\langle T\rangle_r = \underbrace{1\ 2}_{\mathrm{root}(\langle T\rangle_r)}\ 1\ 2\ 1\ 2$

$\langle S\rangle_r = \underbrace{1\ 2}_{\mathrm{root}(\langle S\rangle_r)}\ 1\ 2$

- ω-preorder: $X \preccurlyeq_\omega Y \Leftrightarrow \mathrm{root}(\langle X\rangle_r) = \mathrm{root}(\langle Y\rangle_r) \vee \exists i{:}\, \langle X^\omega[..i]\rangle < \langle Y^\omega[..i]\rangle$

Sorting Conjugates – 2

- adapt approach by [Mantaci+ ‘07] for eBWT
- define the rotational parent distance encoding $\langle T\rangle_r$ of a string T:



$R =$ l y c h e e $\prec_\omega$ $T =$ b a n a n a $=_\omega$ $S =$ k i w i

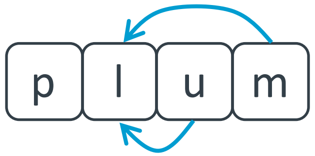
$$\langle R\rangle = \overset{1}{\boxed{\infty}}\overset{2}{\boxed{1}}\overset{3}{\boxed{\infty}}\overset{4}{\boxed{1}}\overset{5}{\boxed{2}}\overset{6}{\boxed{1}} < \langle T\rangle = \overset{1}{\boxed{\infty}}\overset{2}{\boxed{\infty}}\overset{3}{\boxed{1}}\overset{4}{\boxed{2}}\overset{5}{\boxed{1}}\overset{6}{\boxed{2}} \qquad \langle S\rangle_r = \underbrace{\overset{1}{\boxed{1}}\overset{2}{\boxed{2}}}_{\mathrm{root}(\langle S\rangle_r)}\overset{3}{\boxed{1}}\overset{4}{\boxed{2}}$$

- ω-preorder: $X \preccurlyeq_\omega Y \Leftrightarrow \mathrm{root}(\langle X\rangle_r) = \mathrm{root}(\langle Y\rangle_r) \lor \exists i\colon \langle X^\omega[..i]\rangle < \langle Y^\omega[..i]\rangle$

Sorting Conjugates – 3

- $X =_{\omega} Y \Leftrightarrow \langle X^{\omega}[..3z]\rangle = \langle Y^{\omega}[..3z]\rangle$, with $z = \max\{|X|,|Y|\}$, allows for the computation of ω-preorder

	1	2	3		4	5	6	7		8	9	10	11	12
$T =$	f	i	g	·	k	i	w	i	·	a	p	p	l	e



$P =$ p l u m

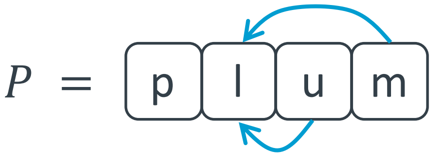
$\langle P\rangle =$ ∞ ∞ 1 2

i	$\mathrm{C}(i)$	$\langle \mathrm{C}(i)^{\omega}[..15]\rangle$
8	apple	∞11345113451134
5	iwik	∞12121212121212
7	ikiw	∞12121212121212
1	fig	∞12312312312312
9	pplea	∞1∞∞∞1134511345
12	eappl	∞∞1134511345113
4	kiwi	∞∞1212121212121
6	wiki	∞∞1212121212121
3	gfi	∞∞1231231231231
11	leapp	∞∞∞113451134511
2	igf	∞∞∞123123123123
10	pleap	∞∞∞∞11345113451

Sorting Conjugates – 3

- $X =_\omega Y \Leftrightarrow \langle X^\omega[..3z]\rangle = \langle Y^\omega[..3z]\rangle$, with $z = \max\{|X|, |Y|\}$, allows for the computation of ω-preorder

	1	2	3		4	5	6	7		8	9	10	11	12
$T =$	f	i	g	·	k	i	w	i	·	a	p	p	l	e



$P =$ p l u m

$\langle P\rangle =$ ∞ ∞ 1 2

i	$\mathrm{C}(i)$	$\langle \mathrm{C}(i)^\omega[..15]\rangle$
8	apple	∞11345113451134
5	iwik	∞12121212121212
7	ikiw	∞12121212121212
1	fig	∞12312312312312
9	pplea	∞1∞∞∞1134511345
12	eappl	∞∞1134511345113
4	kiwi	∞∞1212121212121
6	wiki	∞∞1212121212121
3	gfi	∞∞1231231231231
11	leapp	∞∞∞113451134511
2	igf	∞∞∞123123123123
10	pleap	∞∞∞∞11345113451

Sorting Conjugates – 3

- $X =_\omega Y \Leftrightarrow \langle X^\omega[..3z]\rangle = \langle Y^\omega[..3z]\rangle$, with $z = \max\{|X|, |Y|\}$, allows for the computation of ω-preorder

	1	2	3		4	5	6	7		8	9	10	11	12
$T =$	f	i	g	·	k	i	w	i	·	a	p	p	l	e

$P =$ p l u m

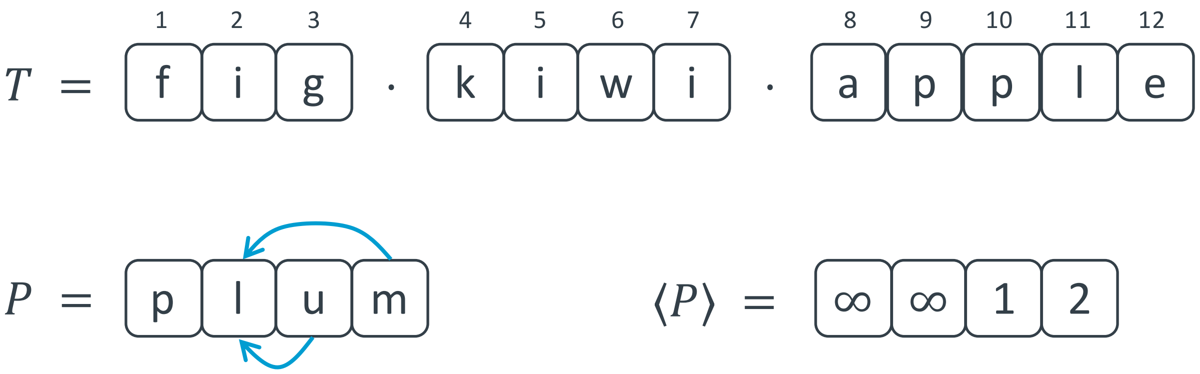
$\langle P \rangle =$ ∞ ∞ 1 2

break ties by index

i	$\mathrm{C}(i)$	$\langle \mathrm{C}(i)^\omega[..15]\rangle$
8	apple	∞11345113451134
5	iwik	∞12121212121212
7	ikiw	∞12121212121212
1	fig	∞12312312312312
9	pplea	∞1∞∞∞1134511345
12	eappl	∞∞1134511345113
4	kiwi	∞∞1212121212121
6	wiki	∞∞1212121212121
3	gfi	∞∞1231231231231
11	leapp	∞∞∞113451134511
2	igf	∞∞∞123123123123
10	pleap	∞∞∞∞11345113451

Sorting Conjugates – 3

- $X =_\omega Y \Leftrightarrow \langle X^\omega[..3z]\rangle = \langle Y^\omega[..3z]\rangle$, with $z = \max\{|X|,|Y|\}$, allows for the computation of ω-preorder
- first column: conjugate array CA



T = f i g · k i w i · a p p l e

P = p l u m

$\langle P\rangle$ = ∞ ∞ 1 2

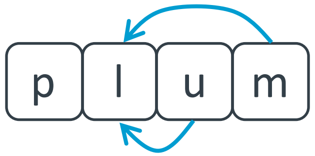
break ties by index

i	$\mathrm{C}(i)$	$\langle \mathrm{C}(i)^\omega[..15]\rangle$
8	apple	∞11345113451134
5	iwik	∞12121212121212
7	ikiw	∞12121212121212
1	fig	∞12312312312312
9	pplea	∞1∞∞∞1134511345
12	eappl	∞∞1134511345113
4	kiwi	∞∞1212121212121
6	wiki	∞∞1212121212121
3	gfi	∞∞1231231231231
11	leapp	∞∞∞113451134511
2	igf	∞∞∞123123123123
10	pleap	∞∞∞∞11345113451

Sorting Conjugates – 3

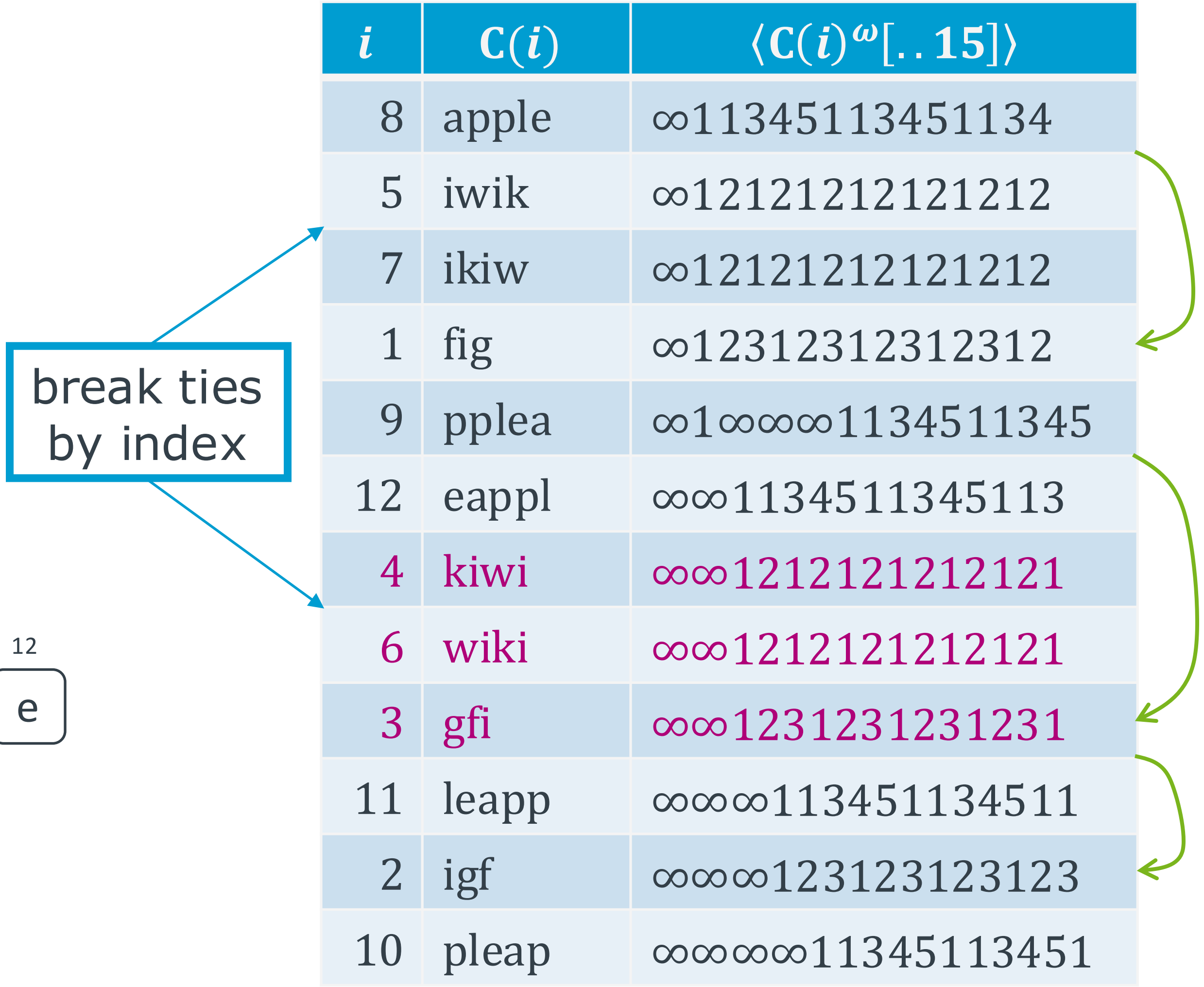
- $X =_\omega Y \Leftrightarrow \langle X^\omega[..3z]\rangle = \langle Y^\omega[..3z]\rangle$, with $z = \max\{|X|, |Y|\}$, allows for the computation of ω-preorder
- first column: conjugate array CA
- red rows: conjugate range $\mathsf{CR}(P)$

	1	2	3		4	5	6	7		8	9	10	11	12
$T =$	f	i	g	·	k	i	w	i	·	a	p	p	l	e



$P =$ p l u m

$\langle P\rangle =$ ∞ ∞ 1 2



break ties by index

i	$\mathsf{C}(i)$	$\langle \mathsf{C}(i)^\omega[..15]\rangle$
8	apple	∞11345113451134
5	iwik	∞12121212121212
7	ikiw	∞12121212121212
1	fig	∞12312312312312
9	pplea	∞1∞∞∞1134511345
12	eappl	∞∞1134511345113
4	kiwi	∞∞1212121212121
6	wiki	∞∞1212121212121
3	gfi	∞∞1231231231231
11	leapp	∞∞∞113451134511
2	igf	∞∞∞123123123123
10	pleap	∞∞∞∞11345113451

FL- and LF-Mapping – 1

- usually maps a conjugates rank to that of its left and right rotation
- due to tie breaks we want to cycle through a text’s encoded roots

FL- and LF-Mapping – 1

- usually maps a conjugates rank to that of its left and right rotation
- due to tie breaks we want to cycle through a text’s encoded roots

$$T_2 = T[4..7] = \boxed{\text{k}}\boxed{\text{i}}\boxed{\text{w}}\boxed{\text{i}}$$

$$\langle T_2 \rangle_r = \underbrace{\boxed{1}\boxed{2}}\underbrace{\boxed{1}\boxed{2}}$$

$$\mathrm{root}(\langle T_2 \rangle_r)$$

FL- and LF-Mapping – 1

- usually maps a conjugates rank to that of its left and right rotation
- due to tie breaks we want to cycle through a text's encoded roots

$T_2 = T[4..7] = \boxed{\text{k}}\boxed{\text{i}}\boxed{\text{w}}\boxed{\text{i}}$

$\langle T_2 \rangle_r = \underbrace{\boxed{1}\boxed{2}}\underbrace{\boxed{1}\boxed{2}}$

$\mathrm{root}(\langle T_2 \rangle_r)$

i	$\mathrm{CA}[i]$	$\mathrm{C}(\mathrm{CA}[i])$	$\mathrm{FL}[i]$	$\mathrm{LF}[i]$
1	8	apple	5	6
2	5	iwik	7	7
3	7	ikiw	8	8
4	1	fig	11	9
5	9	pplea	12	1
6	12	eappl	1	10
7	4	kiwi	2	2
8	6	wiki	3	3
9	3	gfi	4	11
10	11	leapp	6	12
11	2	igf	9	4
12	10	pleap	10	5

FL- and LF-Mapping – 2

- want to represent both mappings space-efficiently
- adapt integer-based representation of [Kim and Cho ‘21]

FL- and LF-Mapping – 2

- want to represent both mappings space-efficiently
- adapt integer-based representation of [Kim and Cho ‘21]

$P =$

1	2	3	4
p	l	u	m

FL- and LF-Mapping – 2

- want to represent both mappings space-efficiently
- adapt integer-based representation of [Kim and Cho ‘21]

$$\begin{array}{ll} \langle P[5..]\rangle = & \\ \langle P[4..]\rangle = & \boxed{\infty} \\ \langle P[3..]\rangle = & \boxed{\infty}\boxed{\infty} \\ \langle P[2..]\rangle = & \boxed{\infty}\boxed{1}\boxed{2} \\ \langle P[1..]\rangle = & \boxed{\infty}\boxed{\infty}\boxed{1}\boxed{2} \end{array}$$

$$[\![P]\!] = \begin{array}{cccc} 1 & 2 & 3 & 4 \\ \boxed{0} & \boxed{2} & \boxed{0} & \boxed{0} \end{array} \qquad P = \begin{array}{cccc} 1 & 2 & 3 & 4 \\ \boxed{\text{p}} & \boxed{\text{l}} & \boxed{\text{u}} & \boxed{\text{m}} \end{array}$$

FL- and LF-Mapping – 2

- want to represent both mappings space-efficiently
- adapt integer-based representation of [Kim and Cho ‘21]

$$\langle P[5..]\rangle =$$

$$\langle P[4..]\rangle = \boxed{\infty}$$

$$\langle P[3..]\rangle = \boxed{\infty}\boxed{\infty}$$

$$\langle P[2..]\rangle = \boxed{\infty}\boxed{1}\boxed{2}$$

$$\langle P[1..]\rangle = \boxed{\infty}\boxed{\infty}\boxed{1}\boxed{2}$$

$$\langle P[1..]\cdot P[..1]\rangle = \boxed{\infty}\boxed{\infty}\boxed{1}\boxed{2}\boxed{1}$$

$$\langle P[4..]\cdot P[..4]\rangle = \boxed{\infty}\boxed{1}\boxed{\infty}\boxed{1}\boxed{2}$$

$$\langle P[3..]\cdot P[..3]\rangle = \boxed{\infty}\boxed{\infty}\boxed{1}\boxed{\infty}\boxed{1}$$

$$\langle P[2..]\cdot P[..2]\rangle = \boxed{\infty}\boxed{1}\boxed{2}\boxed{1}\boxed{4}$$

$$\langle P[1..]\cdot P[..1]\rangle = \boxed{\infty}\boxed{\infty}\boxed{1}\boxed{2}\boxed{1}$$

$$[\![P]\!] = \begin{array}{cccc} 1 & 2 & 3 & 4 \\ \boxed{0} & \boxed{2} & \boxed{0} & \boxed{0} \end{array}$$

$$P = \begin{array}{cccc} 1 & 2 & 3 & 4 \\ \boxed{\text{p}} & \boxed{\text{l}} & \boxed{\text{u}} & \boxed{\text{m}} \end{array}$$

$$[\![P]\!]_r = \begin{array}{cccc} 1 & 2 & 3 & 4 \\ \boxed{0} & \boxed{3} & \boxed{0} & \boxed{1} \end{array}$$

FL- and LF-Mapping – 3

i	$\mathrm{CA}[i]$	$\mathrm{C}(\mathrm{CA}[i])$	$\mathrm{FL}[i]$	$\mathrm{F}[i]$	$[\![\mathrm{C}(\mathrm{CA}[i])]\!]_r$	$\mathrm{L}[i]$	$\mathrm{LF}[i]$
1	8	apple	5	4	41000	0	6
2	5	iwik	7	2	2020	0	7
3	7	ikiw	8	2	2020	0	8
4	1	fig	11	3	300	0	9
5	9	pplea	12	1	10004	4	1
6	12	eappl	1	0	04100	0	10
7	4	kiwi	2	0	0202	2	2
8	6	wiki	3	0	0202	2	3
9	3	gfi	4	0	030	0	11
10	11	leapp	6	0	00410	0	12
11	2	igf	9	0	003	3	4
12	10	pleap	10	0	00041	1	5

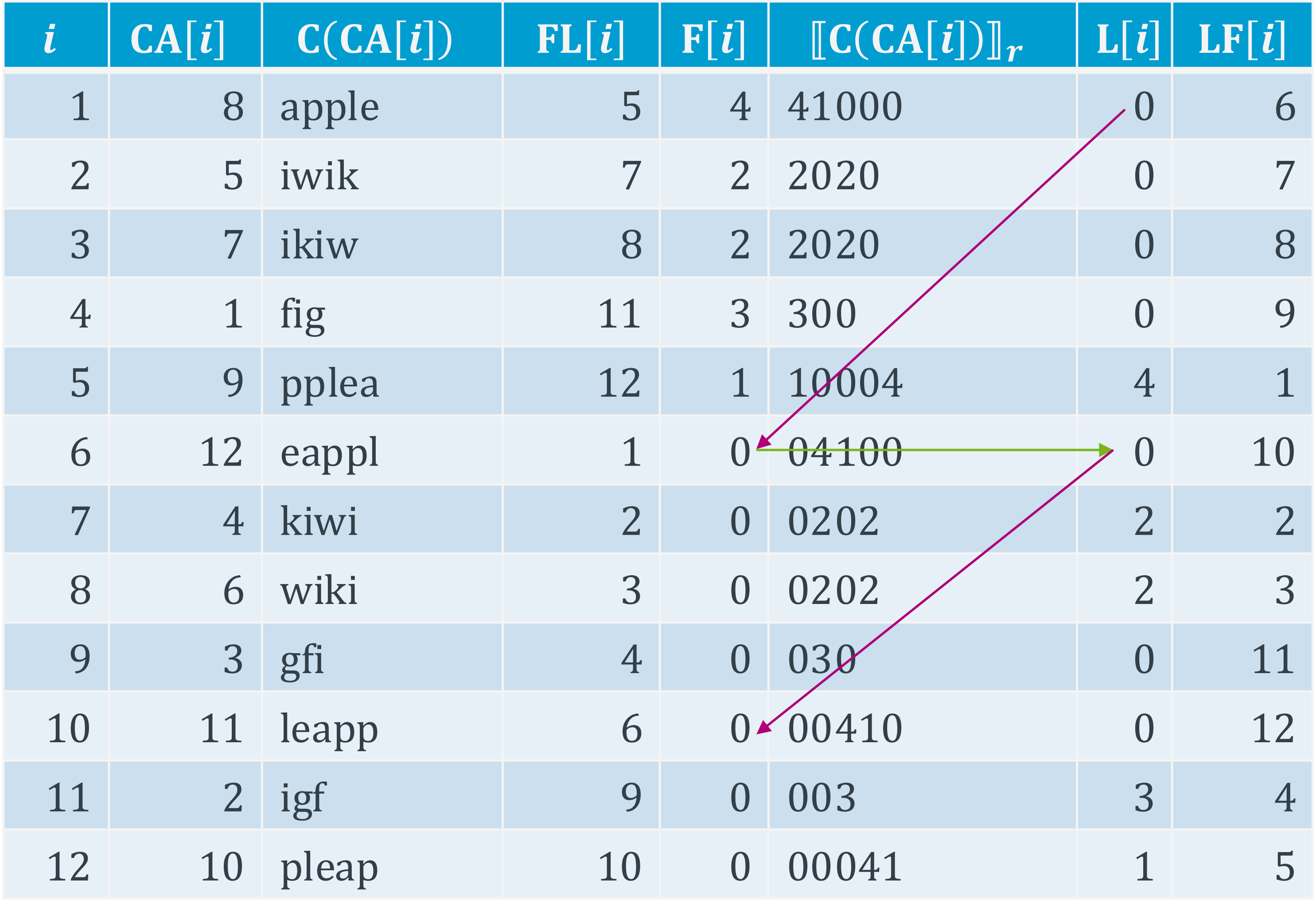
FL- and LF-Mapping – 3

i	$\mathrm{CA}[i]$	$\mathrm{C}(\mathrm{CA}[i])$	$\mathrm{FL}[i]$	$\mathrm{F}[i]$	$[\![\mathrm{C}(\mathrm{CA}[i])]\!]_r$	$\mathrm{L}[i]$	$\mathrm{LF}[i]$
1	8	apple	5	4	41000	0	6
2	5	iwik	7	2	2020	0	7
3	7	ikiw	8	2	2020	0	8
4	1	fig	11	3	300	0	9
5	9	pplea	12	1	10004	4	1
6	12	eappl	1	0	04100	0	10
7	4	kiwi	2	0	0202	2	2
8	6	wiki	3	0	0202	2	3
9	3	gfi	4	0	030	0	11
10	11	leapp	6	0	00410	0	12
11	2	igf	9	0	003	3	4
12	10	pleap	10	0	00041	1	5

FL- and LF-Mapping – 3

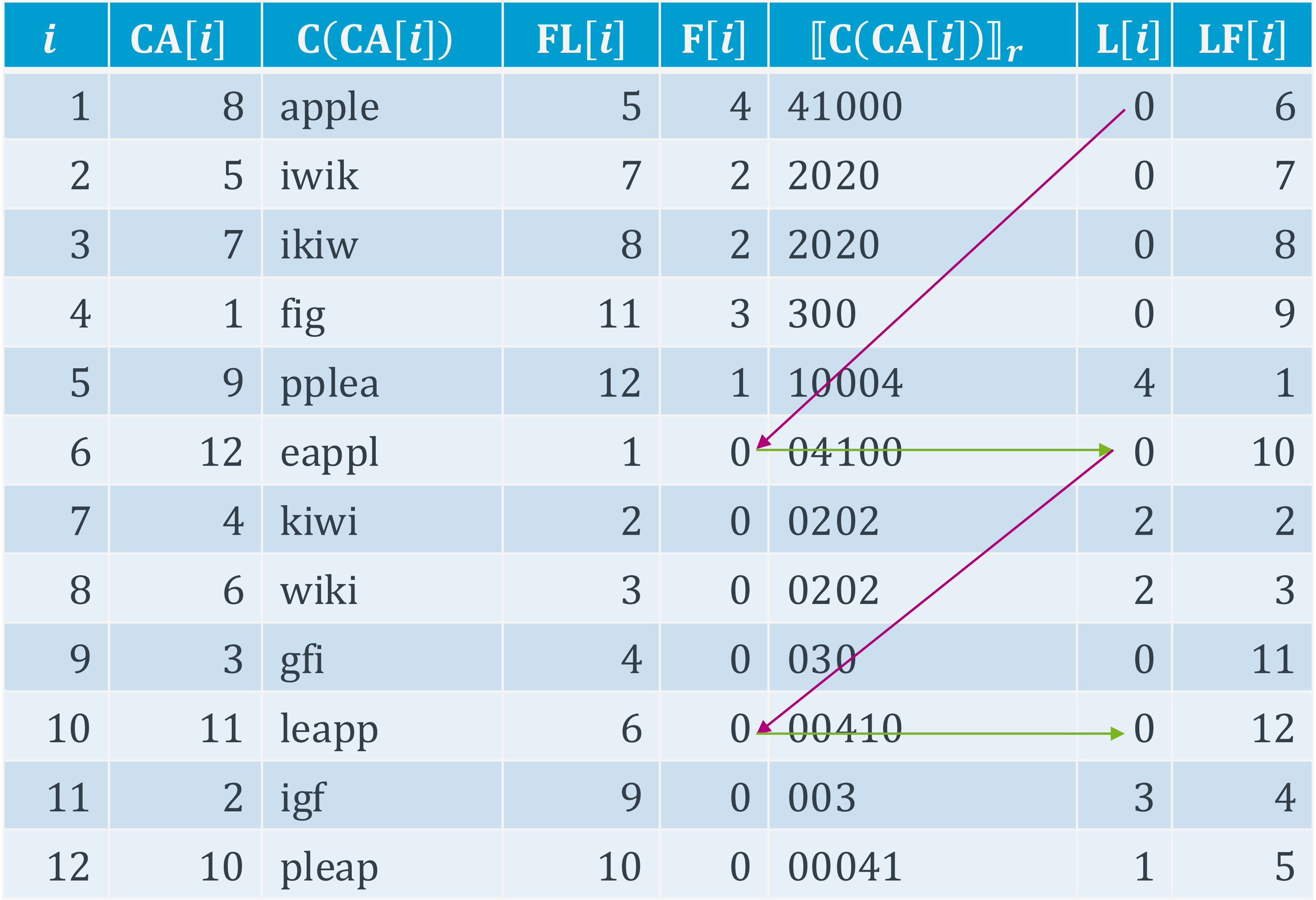
i	$\mathrm{CA}[i]$	$\mathrm{C}(\mathrm{CA}[i])$	$\mathrm{FL}[i]$	$\mathrm{F}[i]$	$[\![\mathrm{C}(\mathrm{CA}[i])]\!]_r$	$\mathrm{L}[i]$	$\mathrm{LF}[i]$
1	8	apple	5	4	41000	0	6
2	5	iwik	7	2	2020	0	7
3	7	ikiw	8	2	2020	0	8
4	1	fig	11	3	300	0	9
5	9	pplea	12	1	10004	4	1
6	12	eappl	1	0	04100	0	10
7	4	kiwi	2	0	0202	2	2
8	6	wiki	3	0	0202	2	3
9	3	gfi	4	0	030	0	11
10	11	leapp	6	0	00410	0	12
11	2	igf	9	0	003	3	4
12	10	pleap	10	0	00041	1	5

FL- and LF-Mapping – 3



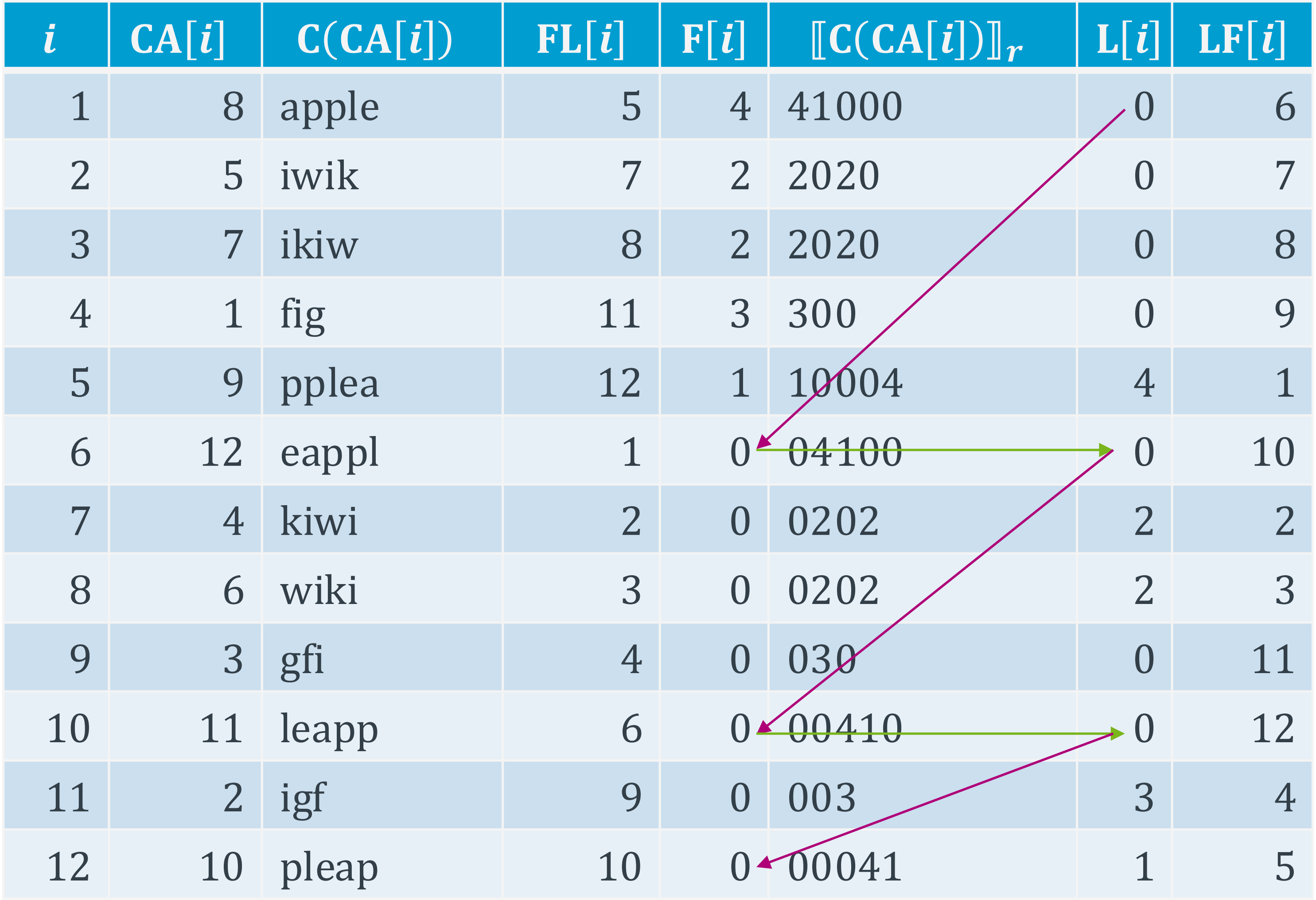
i	$\mathrm{CA}[i]$	$\mathrm{C}(\mathrm{CA}[i])$	$\mathrm{FL}[i]$	$\mathrm{F}[i]$	$\llbracket \mathrm{C}(\mathrm{CA}[i]) \rrbracket_r$	$\mathrm{L}[i]$	$\mathrm{LF}[i]$
1	8	apple	5	4	41000	0	6
2	5	iwik	7	2	2020	0	7
3	7	ikiw	8	2	2020	0	8
4	1	fig	11	3	300	0	9
5	9	pplea	12	1	10004	4	1
6	12	eappl	1	0	04100	0	10
7	4	kiwi	2	0	0202	2	2
8	6	wiki	3	0	0202	2	3
9	3	gfi	4	0	030	0	11
10	11	leapp	6	0	00410	0	12
11	2	igf	9	0	003	3	4
12	10	pleap	10	0	00041	1	5

FL- and LF-Mapping – 3



i	$\mathrm{CA}[i]$	$\mathrm{C}(\mathrm{CA}[i])$	$\mathrm{FL}[i]$	$\mathrm{F}[i]$	$[\![\mathrm{C}(\mathrm{CA}[i])]\!]_r$	$\mathrm{L}[i]$	$\mathrm{LF}[i]$
1	8	apple	5	4	41000	0	6
2	5	iwik	7	2	2020	0	7
3	7	ikiw	8	2	2020	0	8
4	1	fig	11	3	300	0	9
5	9	pplea	12	1	10004	4	1
6	12	eappl	1	0	04100	0	10
7	4	kiwi	2	0	0202	2	2
8	6	wiki	3	0	0202	2	3
9	3	gfi	4	0	030	0	11
10	11	leapp	6	0	00410	0	12
11	2	igf	9	0	003	3	4
12	10	pleap	10	0	00041	1	5

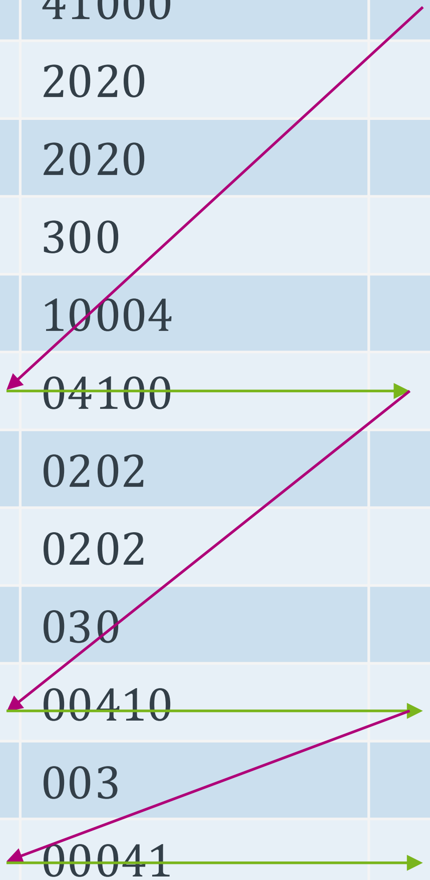
FL- and LF-Mapping – 3



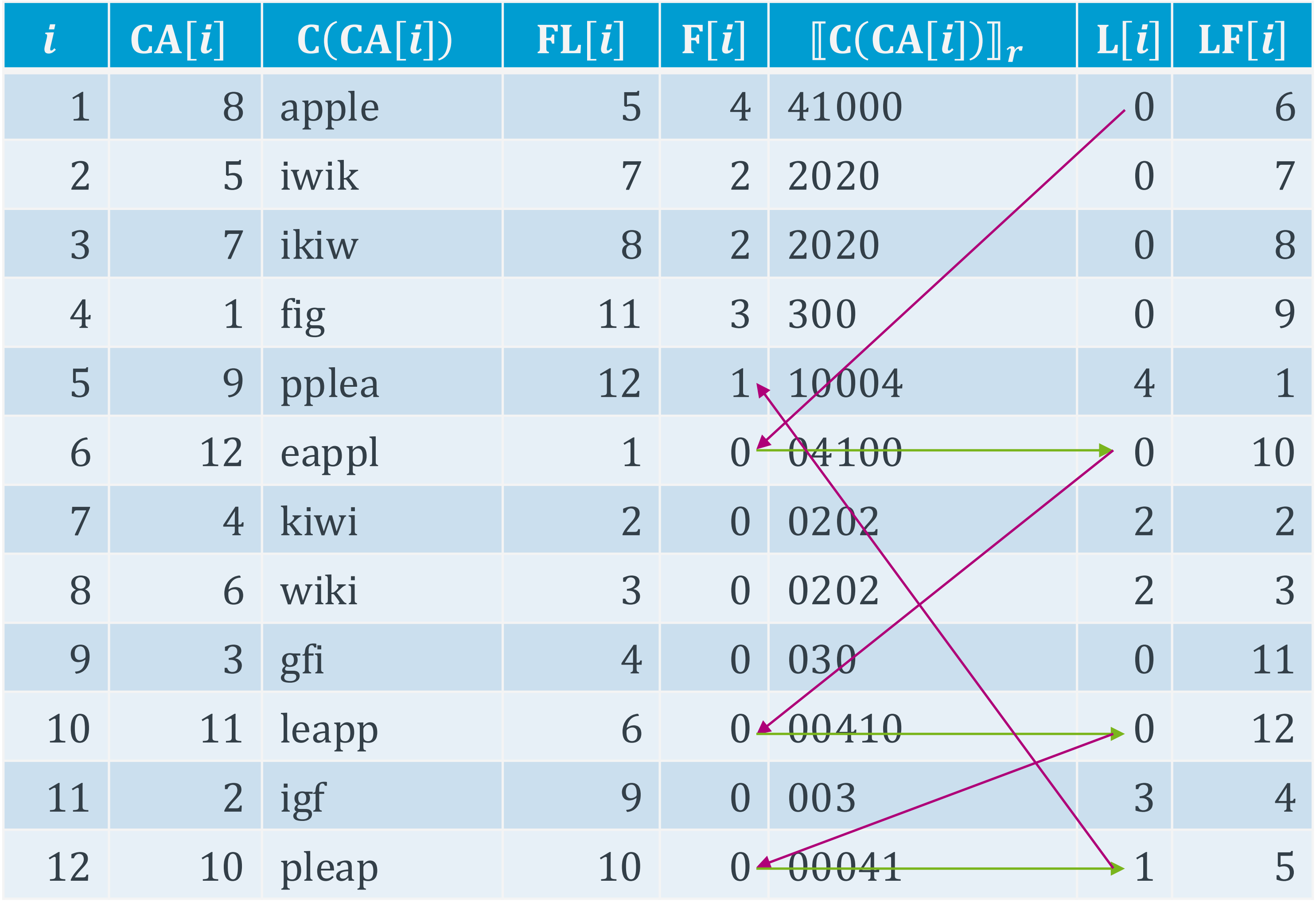
i	$\mathrm{CA}[i]$	$\mathrm{C}(\mathrm{CA}[i])$	$\mathrm{FL}[i]$	$\mathrm{F}[i]$	$[\![\mathrm{C}(\mathrm{CA}[i])]\!]_r$	$\mathrm{L}[i]$	$\mathrm{LF}[i]$
1	8	apple	5	4	41000	0	6
2	5	iwik	7	2	2020	0	7
3	7	ikiw	8	2	2020	0	8
4	1	fig	11	3	300	0	9
5	9	pplea	12	1	10004	4	1
6	12	eappl	1	0	04100	0	10
7	4	kiwi	2	0	0202	2	2
8	6	wiki	3	0	0202	2	3
9	3	gfi	4	0	030	0	11
10	11	leapp	6	0	00410	0	12
11	2	igf	9	0	003	3	4
12	10	pleap	10	0	00041	1	5

FL- and LF-Mapping – 3

i	$\mathrm{CA}[i]$	$\mathrm{C}(\mathrm{CA}[i])$	$\mathrm{FL}[i]$	$\mathrm{F}[i]$	$[\![\mathrm{C}(\mathrm{CA}[i])]\!]_r$	$\mathrm{L}[i]$	$\mathrm{LF}[i]$
1	8	apple	5	4	41000	0	6
2	5	iwik	7	2	2020	0	7
3	7	ikiw	8	2	2020	0	8
4	1	fig	11	3	300	0	9
5	9	pplea	12	1	10004	4	1
6	12	eappl	1	0	04100	0	10
7	4	kiwi	2	0	0202	2	2
8	6	wiki	3	0	0202	2	3
9	3	gfi	4	0	030	0	11
10	11	leapp	6	0	00410	0	12
11	2	igf	9	0	003	3	4
12	10	pleap	10	0	00041	1	5

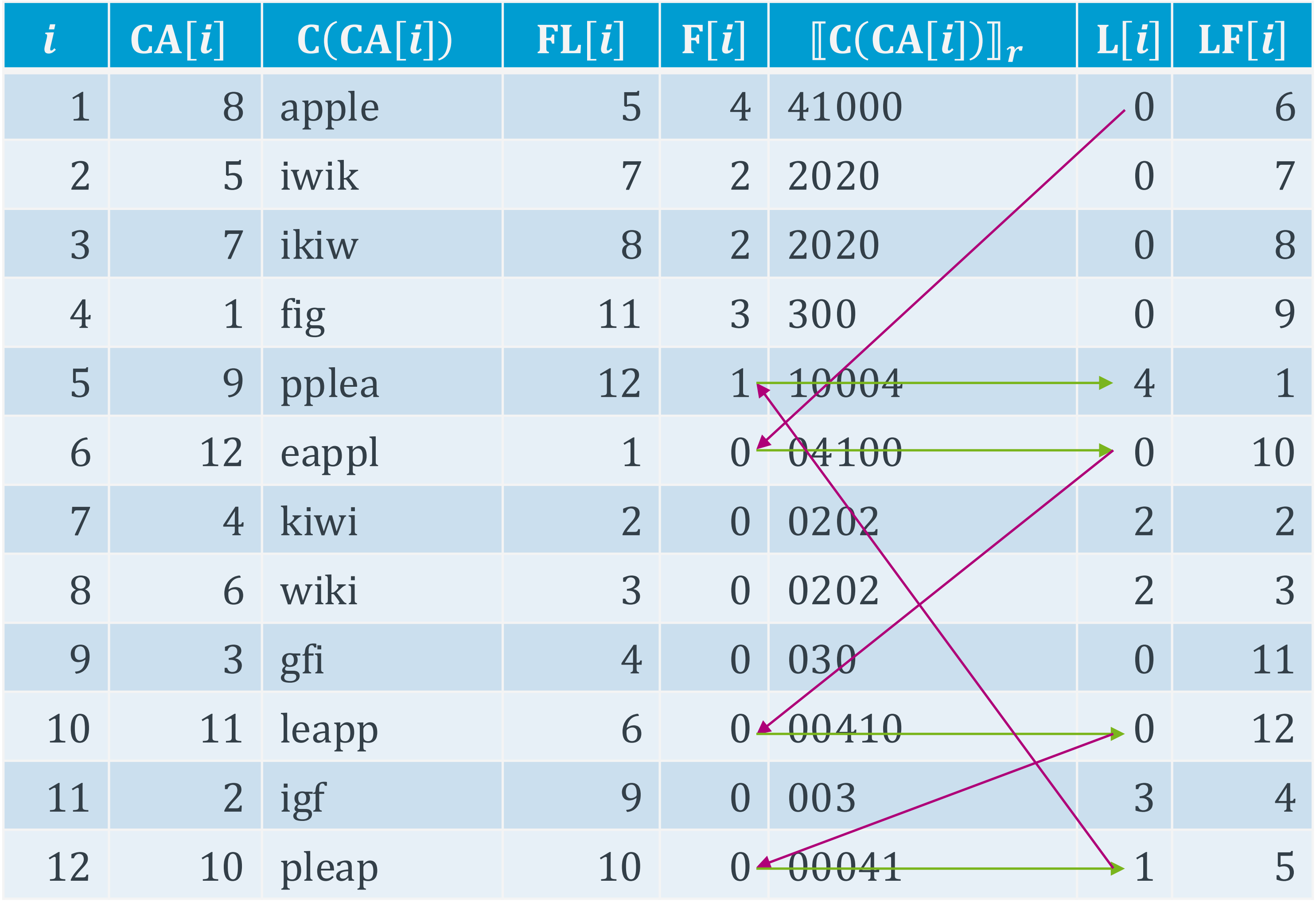


FL- and LF-Mapping – 3



i	$\mathrm{CA}[i]$	$\mathrm{C}(\mathrm{CA}[i])$	$\mathrm{FL}[i]$	$\mathrm{F}[i]$	$[\![\mathrm{C}(\mathrm{CA}[i])]\!]_r$	$\mathrm{L}[i]$	$\mathrm{LF}[i]$
1	8	apple	5	4	41000	0	6
2	5	iwik	7	2	2020	0	7
3	7	ikiw	8	2	2020	0	8
4	1	fig	11	3	300	0	9
5	9	pplea	12	1	10004	4	1
6	12	eappl	1	0	04100	0	10
7	4	kiwi	2	0	0202	2	2
8	6	wiki	3	0	0202	2	3
9	3	gfi	4	0	030	0	11
10	11	leapp	6	0	00410	0	12
11	2	igf	9	0	003	3	4
12	10	pleap	10	0	00041	1	5

FL- and LF-Mapping – 3



i	$\mathrm{CA}[i]$	$\mathrm{C}(\mathrm{CA}[i])$	$\mathrm{FL}[i]$	$\mathrm{F}[i]$	$[\![\mathrm{C}(\mathrm{CA}[i])]\!]_r$	$\mathrm{L}[i]$	$\mathrm{LF}[i]$
1	8	apple	5	4	41000	0	6
2	5	iwik	7	2	2020	0	7
3	7	ikiw	8	2	2020	0	8
4	1	fig	11	3	300	0	9
5	9	pplea	12	1	10004	4	1
6	12	eappl	1	0	04100	0	10
7	4	kiwi	2	0	0202	2	2
8	6	wiki	3	0	0202	2	3
9	3	gfi	4	0	030	0	11
10	11	leapp	6	0	00410	0	12
11	2	igf	9	0	003	3	4
12	10	pleap	10	0	00041	1	5

FL- and LF-Mapping – 3

i	CA[i]	C(CA[i])	FL[i]	F[i]	$[\![\mathrm{C}(\mathrm{CA}[i])]\!]_r$	L[i]	LF[i]
1	8	apple	5	4	41000	0	6
2	5	iwik	7	2	2020	0	7
3	7	ikiw	8	2	2020	0	8
4	1	fig	11	3	300	0	9
5	9	pplea	12	1	10004	4	1
6	12	eappl	1	0	04100	0	10
7	4	kiwi	2	0	0202	2	2
8	6	wiki	3	0	0202	2	3
9	3	gfi	4	0	030	0	11
10	11	leapp	6	0	00410	0	12
11	2	igf	9	0	003	3	4
12	10	pleap	10	0	00041	1	5

FL- and LF-Mapping – 3

i	$\mathrm{CA}[i]$	$\mathrm{C}(\mathrm{CA}[i])$	$\mathrm{FL}[i]$	$\mathrm{F}[i]$	$[\![\mathrm{C}(\mathrm{CA}[i])]\!]_r$	$\mathrm{L}[i]$	$\mathrm{LF}[i]$
1	8	apple	5	4	41000	0	6
2	5	iwik	7	2	2020	0	7
3	7	ikiw	8	2	2020	0	8
4	1	fig	11	3	300	0	9
5	9	pplea	12	1	10004	4	1
6	12	eappl	1	0	04100	0	10
7	4	kiwi	2	0	0202	2	2
8	6	wiki	3	0	0202	2	3
9	3	gfi	4	0	030	0	11
10	11	leapp	6	0	00410	0	12
11	2	igf	9	0	003	3	4
12	10	pleap	10	0	00041	1	5

FL- and LF-Mapping – 3

i	$\mathrm{CA}[i]$	$\mathrm{C}(\mathrm{CA}[i])$	$\mathrm{FL}[i]$	$\mathrm{F}[i]$	$\llbracket \mathrm{C}(\mathrm{CA}[i]) \rrbracket_r$	$\mathrm{L}[i]$	$\mathrm{LF}[i]$
1	8	apple	5	4	41000	0	6
2	5	iwik	7	2	2020	0	7
3	7	ikiw	8	2	2020	0	8
4	1	fig	11	3	300	0	9
5	9	pplea	12	1	10004	4	1
6	12	eappl	1	0	04100	0	10
7	4	kiwi	2	0	0202	2	2
8	6	wiki	3	0	0202	2	3
9	3	gfi	4	0	030	0	11
10	11	leapp	6	0	00410	0	12
11	2	igf	9	0	003	3	4
12	10	pleap	10	0	00041	1	5

FL- and LF-Mapping – 3

i	$\mathrm{CA}[i]$	$\mathrm{C}(\mathrm{CA}[i])$	$\mathrm{FL}[i]$	$\mathrm{F}[i]$	$[\![\mathrm{C}(\mathrm{CA}[i])]\!]_r$	$\mathrm{L}[i]$	$\mathrm{LF}[i]$
1	8	apple	5	4	41000	0	6
2	5	iwik	7	2	2020	0	7
3	7	ikiw	8	2	2020	0	8
4	1	fig	11	3	300	0	9
5	9	pplea	12	1	10004	4	1
6	12	eappl	1	0	04100	0	10
7	4	kiwi	2	0	0202	2	2
8	6	wiki	3	0	0202	2	3
9	3	gfi	4	0	030	0	11
10	11	leapp	6	0	00410	0	12
11	2	igf	9	0	003	3	4
12	10	pleap	10	0	00041	1	5

FL- and LF-Mapping – 3

i	CA[i]	C(CA[i])	FL[i]	F[i]	$[\![\mathrm{C}(\mathrm{CA}[i])]\!]_r$	L[i]	LF[i]
1	8	apple	5	4	41000	0	6
2	5	iwik	7	2	2020	0	7
3	7	ikiw	8	2	2020	0	8
4	1	fig	11	3	300	0	9
5	9	pplea	12	1	10004	4	1
6	12	eappl	1	0	04100	0	10
7	4	kiwi	2	0	0202	2	2
8	6	wiki	3	0	0202	2	3
9	3	gfi	4	0	030	0	11
10	11	leapp	6	0	00410	0	12
11	2	igf	9	0	003	3	4
12	10	pleap	10	0	00041	1	5

FL- and LF-Mapping – 4

- e_k: number of occurrences of ∞ in $\langle P[k..]\rangle$
- $j \in \mathrm{CR}(P[k+1..])$
- if $e_k > 1$ then $LF[j] \in \mathrm{CR}(P[k..]) \Leftrightarrow L[j] = [\![P]\!][k]$
- if $e_k = 1$ then $LF[j] \in \mathrm{CR}(P[k..]) \Leftrightarrow L[j] \geq [\![P]\!][k]$

i	$\mathrm{CA}[i]$	$\mathrm{F}[i]$	$\mathrm{L}[i]$
1	8	4	0
2	5	2	0
3	7	2	0
4	1	3	0
5	9	1	4
6	12	0	0
7	4	0	2
8	6	0	2
9	3	0	0
10	11	0	0
11	2	0	3
12	10	0	1

FL- and LF-Mapping – 4

- e_k: number of occurrences of ∞ in $\langle P[k..]\rangle$
- $j \in \mathrm{CR}(P[k+1..])$
- if $e_k > 1$ then $LF[j] \in \mathrm{CR}(P[k..]) \Leftrightarrow L[j] = [\![P]\!][k]$
- if $e_k = 1$ then $LF[j] \in \mathrm{CR}(P[k..]) \Leftrightarrow L[j] \geq [\![P]\!][k]$

	1	2	3	4
P =	p	l	u	m

	1	2	3	4
$[\![P]\!]$ =	0	2	0	0

i	CA[i]	F[i]	L[i]
1	8	4	0
2	5	2	0
3	7	2	0
4	1	3	0
5	9	1	4
6	12	0	0
7	4	0	2
8	6	0	2
9	3	0	0
10	11	0	0
11	2	0	3
12	10	0	1

FL- and LF-Mapping – 4

- e_k: number of occurrences of ∞ in $\langle P[k..]\rangle$
- $j \in \mathrm{CR}(P[k+1..])$
- if $e_k > 1$ then $LF[j] \in \mathrm{CR}(P[k..]) \Leftrightarrow L[j] = [\![P]\!][k]$
- if $e_k = 1$ then $LF[j] \in \mathrm{CR}(P[k..]) \Leftrightarrow L[j] \geq [\![P]\!][k]$

	1	2	3	4
$P =$	p	l	u	m
$[\![P]\!] =$	0	2	0	0

k	e_k	$[\![P]\!][k]$	$CR(P[k+1..])$
4	1	0	$[1..12]$
3	2	0	
2	1	2	
1	2	0	

i	$\mathrm{CA}[i]$	$\mathrm{F}[i]$	$\mathrm{L}[i]$
1	8	4	0
2	5	2	0
3	7	2	0
4	1	3	0
5	9	1	4
6	12	0	0
7	4	0	2
8	6	0	2
9	3	0	0
10	11	0	0
11	2	0	3
12	10	0	1

FL- and LF-Mapping – 4

- e_k: number of occurrences of ∞ in $\langle P[k..]\rangle$
- $j \in \mathrm{CR}(P[k+1..])$
- if $e_k > 1$ then $LF[j] \in \mathrm{CR}(P[k..]) \Leftrightarrow L[j] = \llbracket P \rrbracket[k]$
- if $e_k = 1$ then $LF[j] \in \mathrm{CR}(P[k..]) \Leftrightarrow L[j] \geq \llbracket P \rrbracket[k]$

	1	2	3	4
$P =$	p	l	u	m

	1	2	3	4
$\llbracket P \rrbracket =$	0	2	0	0

k	e_k	$\llbracket P \rrbracket[k]$	$CR(P[k+1..])$
4	1	0	$[1..12]$
3	2	0	$[1..12]$
2	1	2	
1	2	0	

i	CA$[i]$	F$[i]$	L$[i]$
1	8	4	0
2	5	2	0
3	7	2	0
4	1	3	0
5	9	1	4
6	12	0	0
7	4	0	2
8	6	0	2
9	3	0	0
10	11	0	0
11	2	0	3
12	10	0	1

FL- and LF-Mapping – 4

- e_k: number of occurrences of ∞ in $\langle P[k..]\rangle$
- $j \in \mathrm{CR}(P[k+1..])$
- if $e_k > 1$ then $LF[j] \in \mathrm{CR}(P[k..]) \Leftrightarrow L[j] = [\![P]\!][k]$
- if $e_k = 1$ then $LF[j] \in \mathrm{CR}(P[k..]) \Leftrightarrow L[j] \geq [\![P]\!][k]$

$$P = \begin{array}{cccc} 1 & 2 & 3 & 4 \\ \boxed{\mathrm{p}} & \boxed{\mathrm{l}} & \boxed{\mathrm{u}} & \boxed{\mathrm{m}} \end{array}$$

$$[\![P]\!] = \begin{array}{cccc} 1 & 2 & 3 & 4 \\ \boxed{0} & \boxed{2} & \boxed{0} & \boxed{0} \end{array}$$

k	e_k	$[\![P]\!][k]$	$CR(P[k+1..])$
4	1	0	$[1..12]$
3	2	0	$[1..12]$
2	1	2	
1	2	0	

i	$\mathrm{CA}[i]$	$\mathrm{F}[i]$	$\mathrm{L}[i]$
1	8	4	0
2	5	2	0
3	7	2	0
4	1	3	0
5	9	1	4
6	12	0	0
7	4	0	2
8	6	0	2
9	3	0	0
10	11	0	0
11	2	0	3
12	10	0	1

FL- and LF-Mapping – 4

- e_k: number of occurrences of ∞ in $\langle P[k..]\rangle$
- $j \in \mathrm{CR}(P[k+1..])$
- if $e_k > 1$ then $LF[j] \in \mathrm{CR}(P[k..]) \Leftrightarrow L[j] = [\![P]\!][k]$
- if $e_k = 1$ then $LF[j] \in \mathrm{CR}(P[k..]) \Leftrightarrow L[j] \geq [\![P]\!][k]$

	1	2	3	4
P =	p	l	u	m

	1	2	3	4
$[\![P]\!]$ =	0	2	0	0

k	e_k	$[\![P]\!][k]$	$CR(P[k+1..])$
4	1	0	$[1..12]$
3	2	0	$[1..12]$
2	1	2	$[6..12]$
1	2	0	

i	CA[i]	F[i]	L[i]
1	8	4	0
2	5	2	0
3	7	2	0
4	1	3	0
5	9	1	4
6	12	0	0
7	4	0	2
8	6	0	2
9	3	0	0
10	11	0	0
11	2	0	3
12	10	0	1

FL- and LF-Mapping – 4

- e_k: number of occurrences of ∞ in $\langle P[k..]\rangle$
- $j \in \mathrm{CR}(P[k+1..])$
- if $e_k > 1$ then $LF[j] \in \mathrm{CR}(P[k..]) \Leftrightarrow L[j] = [\![P]\!][k]$
- if $e_k = 1$ then $LF[j] \in \mathrm{CR}(P[k..]) \Leftrightarrow L[j] \geq [\![P]\!][k]$

	1	2	3	4
$P =$	p	l	u	m

	1	2	3	4
$[\![P]\!] =$	0	2	0	0

k	e_k	$[\![P]\!][k]$	$CR(P[k+1..])$
4	1	0	$[1..12]$
3	2	0	$[1..12]$
2	1	2	$[6..12]$
1	2	0	

i	$\mathrm{CA}[i]$	$\mathrm{F}[i]$	$\mathrm{L}[i]$
1	8	4	0
2	5	2	0
3	7	2	0
4	1	3	0
5	9	1	4
6	12	0	0
7	4	0	2
8	6	0	2
9	3	0	0
10	11	0	0
11	2	0	3
12	10	0	1

FL- and LF-Mapping – 4

- e_k: number of occurrences of ∞ in $\langle P[k..]\rangle$
- $j \in \mathrm{CR}(P[k+1..])$
- if $e_k > 1$ then $LF[j] \in \mathrm{CR}(P[k..]) \Leftrightarrow L[j] = [\![P]\!][k]$
- if $e_k = 1$ then $LF[j] \in \mathrm{CR}(P[k..]) \Leftrightarrow L[j] \geq [\![P]\!][k]$

$P =$

1	2	3	4
p	l	u	m

$[\![P]\!] =$

1	2	3	4
0	2	0	0

k	e_k	$[\![P]\!][k]$	$CR(P[k+1..])$
4	1	0	$[1..12]$
3	2	0	$[1..12]$
2	1	2	$[6..12]$
1	2	0	$[2..4]$

i	$\mathrm{CA}[i]$	$\mathrm{F}[i]$	$\mathrm{L}[i]$
1	8	4	0
2	5	2	0
3	7	2	0
4	1	3	0
5	9	1	4
6	12	0	0
7	4	0	2
8	6	0	2
9	3	0	0
10	11	0	0
11	2	0	3
12	10	0	1

FL- and LF-Mapping – 4

- e_k: number of occurrences of ∞ in $\langle P[k..] \rangle$
- $j \in \mathrm{CR}(P[k+1..])$
- if $e_k > 1$ then $LF[j] \in \mathrm{CR}(P[k..]) \Leftrightarrow L[j] = \llbracket P \rrbracket[k]$
- if $e_k = 1$ then $LF[j] \in \mathrm{CR}(P[k..]) \Leftrightarrow L[j] \geq \llbracket P \rrbracket[k]$

	1	2	3	4
$P =$	p	l	u	m

	1	2	3	4
$\llbracket P \rrbracket =$	0	2	0	0

k	e_k	$\llbracket P \rrbracket[k]$	$CR(P[k+1..])$
4	1	0	$[1..12]$
3	2	0	$[1..12]$
2	1	2	$[6..12]$
1	2	0	$[2..4]$

i	CA$[i]$	F$[i]$	L$[i]$
1	8	4	0
2	5	2	0
3	7	2	0
4	1	3	0
5	9	1	4
6	12	0	0
7	4	0	2
8	6	0	2
9	3	0	0
10	11	0	0
11	2	0	3
12	10	0	1

FL- and LF-Mapping – 4

- e_k: number of occurrences of ∞ in $\langle P[k..]\rangle$
- $j \in \mathrm{CR}(P[k+1..])$
- if $e_k > 1$ then $LF[j] \in \mathrm{CR}(P[k..]) \Leftrightarrow L[j] = [\![P]\!][k]$
- if $e_k = 1$ then $LF[j] \in \mathrm{CR}(P[k..]) \Leftrightarrow L[j] \geq [\![P]\!][k]$

P =

1	2	3	4
p	l	u	m

$[\![P]\!]$ =

1	2	3	4
0	2	0	0

k	e_k	$[\![P]\!][k]$	$CR(P[k+1..])$
4	1	0	$[1..12]$
3	2	0	$[1..12]$
2	1	2	$[6..12]$
1	2	0	$[2..4]$

i	CA$[i]$	F$[i]$	L$[i]$
1	8	4	0
2	5	2	0
3	7	2	0
4	1	3	0
5	9	1	4
6	12	0	0
7	4	0	2
8	6	0	2
9	3	0	0
10	11	0	0
11	2	0	3
12	10	0	1

Then What?

- update conjugate ranges time-efficiently (independent of text length n)
- resulting index (cBWT) has dynamic and static variants
  - static variant is straightforward adaptation of index by [Kim and Cho ‘21]
  - dynamic variant uses a modified LCP-array by [Hashimoto+ ‘22, Iseri+ ‘24]
- at the core of the construction algorithm:
  - construction of single text index
  - extension of existing index by another text
- construction adapts techniques by [Hashimoto+ ‘22, Iseri+ ‘24]

Summary & Future Work

- cBWT index:
  - solves CTPM to multiple and (optionally) circular texts
  - dynamic variant: $O(n \log \sigma)$ bits of space and CTPM in $O(mt)$ time
  - static variant: $3n + o(n)$ bits of space and CTPM in $O(m)$ time
  - both variants constructed in $O(n \log \sigma)$ bits of space and $O(nt)$ time
- future work:
  - less space
  - implementation

$$\boxed{t = \frac{\log \sigma \log n}{\log \log n}}$$