FM Index, part 2

Ben Langmead

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Department of Computer Science

Indexing

Full-text index answers queries:

Locate(P), where $P \in \Sigma^m$, returns all offsets where P matches a substring of T

Count(P) returns # of offsets where P matches a substring of T

Extract(i, m) returns $T[i : i + m - 1]$ (length-m substring starting at i)

FM Index: querying

$ a b a a b **a**
a $ a b a a **b**
a a b a $ a **b**
a b a $ a b **a**
a b a a b a **$**
b a $ a b a **a**
b a a b a $ **a**

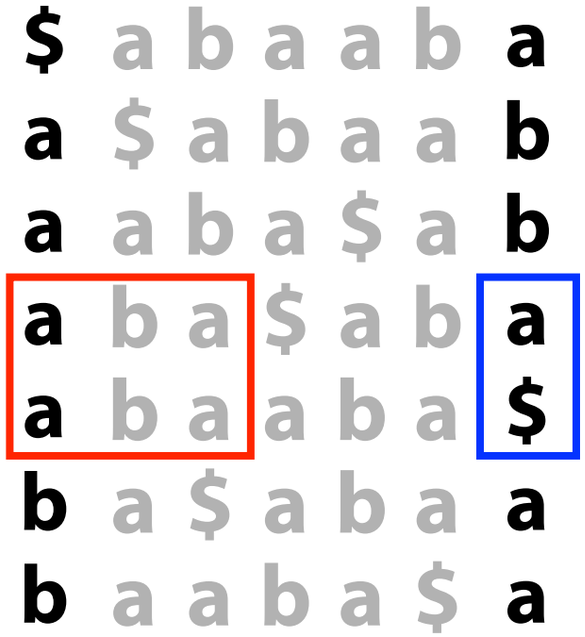
FM Index: querying

Rows with same prefix are consecutive

$ a b a a b a
a $ a b a a b
a a b a $ a b
a b a $ a b a
a b a a b a $
b a $ a b a a
b a a b a $ a

FM Index: querying

Rows with same prefix are consecutive



Characters in last column are those *preceding* the prefixes (to their *left* in T)

FM Index: querying

F						*L*
$	a	b	a	a	b	a_0
a_0	$	a	b	a	a	b_0
a_1	a	b	a	$	a	b_1
a_2	b	a	$	a	b	a_1
a_3	b	a	a	b	a	$
b_0	a	$	a	b	a	a_2
b_1	a	a	b	a	$	a_3

FM Index: querying

Subscripts are ranks in L

FM Index: querying

Given pattern P, $|P| = m$, start with shortest suffix of and match successively longer suffixes

$P =$ **aba**

F	L
$ a b a a b a$	a_0
$a_0 $ a b a a b_0	
a_1 a b a $ a b_1	
a_2 b a $ a b a_1	
a_3 b a a b a $	
b_0 a $ a b a a_2	
b_1 a a b a $ a_3	

FM Index: querying

Given pattern P, $|P| = m$, start with shortest suffix of and match successively longer suffixes

$P =$ **aba**

Easy to find all the rows beginning with **a**

$[C[\mathrm{a}], C[\mathrm{b}]) = [1,5)$

F						L
$	a	b	a	a	b	**a**$_0$
a$_0$	$	a	b	a	a	**b**$_0$
a$_1$	a	b	a	$	a	**b**$_1$
a$_2$	b	a	$	a	b	**a**$_1$
a$_3$	b	a	a	b	a	**$**
b$_0$	a	$	a	b	a	**a**$_2$
b$_1$	a	a	b	a	$	**a**$_3$

FM Index: querying

We have rows beginning with **a**, now we want rows beginning with **ba**

$P =$ **aba**

F						*L*
$	a	b	a	a	b	**a**$_0$
a$_0$	$	a	b	a	a	**b**$_0$
a$_1$	a	b	a	$	a	**b**$_1$
a$_2$	b	a	$	a	b	**a**$_1$
a$_3$	b	a	a	b	a	**$**
b$_0$	a	$	a	b	a	**a**$_2$
b$_1$	a	a	b	a	$	**a**$_3$

FM Index: querying

We have rows beginning with **a**, now we want rows beginning with **ba**

$P =$ **aba**

F						*L*
$	a	b	a	a	b	**a_0**
a_0	**$**	a	b	a	a	**b_0**
a_1	a	b	a	**$**	a	**b_1**
a_2	b	a	**$**	a	b	**a_1**
a_3	b	a	a	b	a	**$**
b_0	a	**$**	a	b	a	**a_2**
b_1	a	a	b	a	**$**	**a_3**

← Look at those rows in *L*.

b_0, **b_1** are **b**s occurring just to left.

FM Index: querying

We have rows beginning with **a**, now we want rows beginning with **ba**

$P =$ **aba**

F						*L*
$	a	b	a	a	b	**$a**$_0$
a$_0$	$	a	b	a	a	**b**$_0$
a$_1$	a	b	a	$	a	**b**$_1$
a$_2$	b	a	$	a	b	**a**$_1$
a$_3$	b	a	a	b	a	**$**
b$_0$	a	$	a	b	a	**a**$_2$
b$_1$	a	a	b	a	$	**a**$_3$

← Look at those rows in *L*.
b$_0$, **b**$_1$ are **b**s occurring just to left.

$P =$ **aba**

F						*L*
$	a	b	a	a	b	**$a**$_0$
a$_0$	$	a	b	a	a	**b**$_0$
a$_1$	a	b	a	$	a	**b**$_1$
a$_2$	b	a	$	a	b	**a**$_1$
a$_3$	b	a	a	b	a	**$**
b$_0$	a	$	a	b	a	**a**$_2$
b$_1$	a	a	b	a	$	**a**$_3$

FM Index: querying

We have rows beginning with **a**, now we want rows beginning with **ba**

$P =$ **aba**

F						*L*
$	a	b	a	a	b	**a**$_0$
a $_0$	**$**	a	b	a	a	**b**$_0$
a $_1$	a	b	a	**$**	a	**b**$_1$
a $_2$	b	a	**$**	a	b	**a**$_1$
a $_3$	b	a	a	b	a	**$**
b $_0$	a	**$**	a	b	a	**a**$_2$
b $_1$	a	a	b	a	**$**	**a**$_3$

← Look at those rows in *L*.
b $_0$, **b**$_1$ are **b**s occurring just to left.

Use LF Mapping. Let new range delimit those **b**s →

$P =$ **aba**

F						*L*
$	a	b	a	a	b	**a**$_0$
a $_0$	**$**	a	b	a	a	**b**$_0$
a $_1$	a	b	a	**$**	a	**b**$_1$
a $_2$	b	a	**$**	a	b	**a**$_1$
a $_3$	b	a	a	b	a	**$**
b $_0$	a	**$**	a	b	a	**a**$_2$
b $_1$	a	a	b	a	**$**	**a**$_3$

FM Index: querying

We have rows beginning with **ba**, now we seek rows beginning with **aba**

$P =$ **aba**

F						*L*
$	a	b	a	a	b	**a**$_0$
a$_0$	$	a	b	a	a	**b**$_0$
a$_1$	a	b	a	$	a	**b**$_1$
a$_2$	b	a	$	a	b	**a**$_1$
a$_3$	b	a	a	b	a	**$**
b$_0$	a	$	a	b	a	**a**$_2$
b$_1$	a	a	b	a	$	**a**$_3$

FM Index: querying

We have rows beginning with **ba**, now we seek rows beginning with **aba**

$P =$ **aba**

F						*L*
$	a	b	a	a	b	$\mathbf{a_0}$
$\mathbf{a_0}$	$	a	b	a	a	$\mathbf{b_0}$
$\mathbf{a_1}$	a	b	a	$	a	$\mathbf{b_1}$
$\mathbf{a_2}$	b	a	$	a	b	$\mathbf{a_1}$
$\mathbf{a_3}$	b	a	a	b	a	**$**
$\mathbf{b_0}$	a	$	a	b	a	$\mathbf{a_2}$
$\mathbf{b_1}$	a	a	b	a	$	$\mathbf{a_3}$

← $\mathbf{a_2}$, $\mathbf{a_3}$ occur just to left.

FM Index: querying

We have rows beginning with **ba**, now we seek rows beginning with **aba**

$P =$ **aba**

F						*L*
$	a	b	a	a	b	$\mathbf{a_0}$
$\mathbf{a_0}$	$	a	b	a	a	$\mathbf{b_0}$
$\mathbf{a_1}$	a	b	a	$	a	$\mathbf{b_1}$
$\mathbf{a_2}$	b	a	$	a	b	$\mathbf{a_1}$
$\mathbf{a_3}$	b	a	a	b	a	**$**
$\mathbf{b_0}$	a	$	a	b	a	$\mathbf{a_2}$
$\mathbf{b_1}$	a	a	b	a	$	$\mathbf{a_3}$

← $\mathbf{a_2}$, $\mathbf{a_3}$ occur just to left.

Use LF Mapping →

$P =$ **aba**

F						*L*
$	a	b	a	a	b	$\mathbf{a_0}$
$\mathbf{a_0}$	$	a	b	a	a	$\mathbf{b_0}$
$\mathbf{a_1}$	a	b	a	$	a	$\mathbf{b_1}$
$\mathbf{a_2}$	b	a	$	a	b	$\mathbf{a_1}$
$\mathbf{a_3}$	b	a	a	b	a	**$**
$\mathbf{b_0}$	a	$	a	b	a	$\mathbf{a_2}$
$\mathbf{b_1}$	a	a	b	a	$	$\mathbf{a_3}$

FM Index: querying

We have rows beginning with **ba**, now we seek rows beginning with **aba**

$P =$ **aba**

F						*L*
$	a	b	a	a	b	$\mathbf{a_0}$
$\mathbf{a_0}$	$	a	b	a	a	$\mathbf{b_0}$
$\mathbf{a_1}$	a	b	a	$	a	$\mathbf{b_1}$
$\mathbf{a_2}$	b	a	$	a	b	$\mathbf{a_1}$
$\mathbf{a_3}$	b	a	a	b	a	**$**
$\mathbf{b_0}$	a	$	a	b	a	$\mathbf{a_2}$
$\mathbf{b_1}$	a	a	b	a	$	$\mathbf{a_3}$

← $\mathbf{a_2}$, $\mathbf{a_3}$ occur just to left.

Use LF Mapping →

$P =$ **aba**

F						*L*
$	a	b	a	a	b	$\mathbf{a_0}$
$\mathbf{a_0}$	$	a	b	a	a	$\mathbf{b_0}$
$\mathbf{a_1}$	a	b	a	$	a	$\mathbf{b_1}$
$\mathbf{a_2}$	b	a	$	a	b	$\mathbf{a_1}$
$\mathbf{a_3}$	b	a	a	b	a	**$**
$\mathbf{b_0}$	a	$	a	b	a	$\mathbf{a_2}$
$\mathbf{b_1}$	a	a	b	a	$	$\mathbf{a_3}$

Now we have the rows with prefix **aba**

$$T.\text{count}(\text{aba}) = 2$$

FM Index: querying

When *P* does not occur in *T*, we eventually fail to find next character in *L*:

$P =$ **bba**

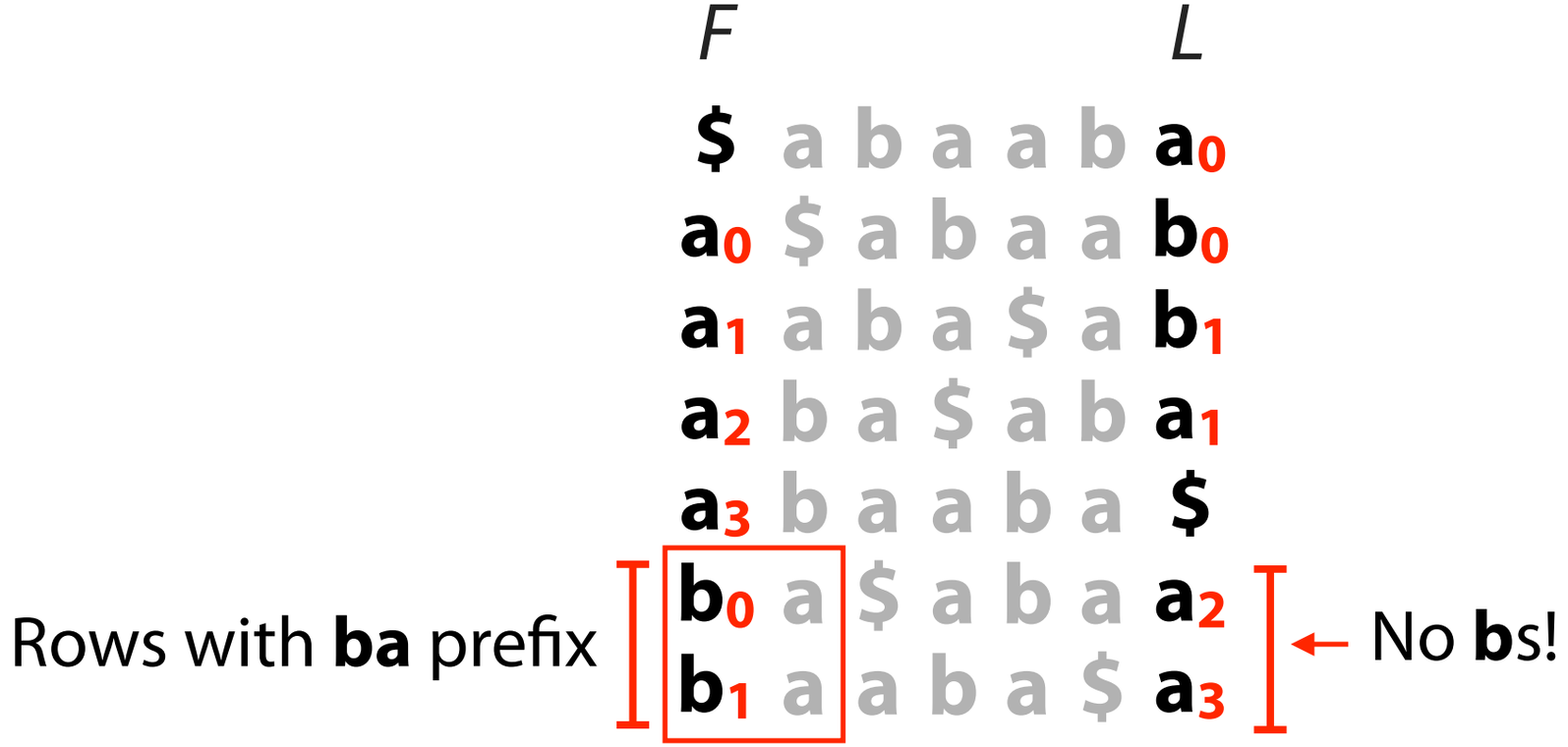
F						*L*
$	a	b	a	a	b	$\mathbf{a_0}$
$\mathbf{a_0}$	$	a	b	a	a	$\mathbf{b_0}$
$\mathbf{a_1}$	a	b	a	$	a	$\mathbf{b_1}$
$\mathbf{a_2}$	b	a	$	a	b	$\mathbf{a_1}$
$\mathbf{a_3}$	b	a	a	b	a	**$**
$\mathbf{b_0}$	a	$	a	b	a	$\mathbf{a_2}$
$\mathbf{b_1}$	a	a	b	a	$	$\mathbf{a_3}$

Rows with **ba** prefix

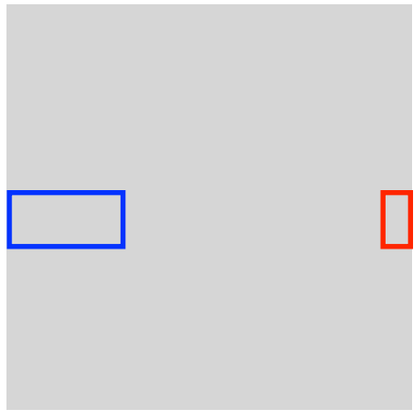
FM Index: querying

When *P* does not occur in *T*, we eventually fail to find next character in *L*:

$$P = \mathbf{bba}$$



FM Index: querying



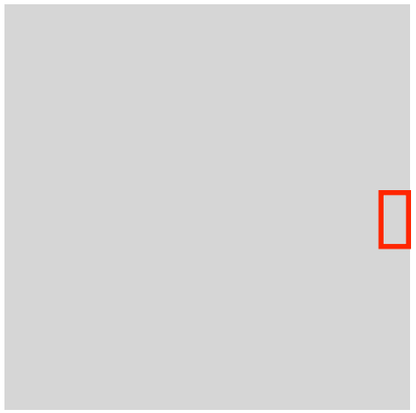
Colors show what parts of matrix are shown on right

final char (L)	sorted rotations
a	n to decompress. It achieves compression
o	n to perform only comparisons to a depth
o	n transformation} This section describes
o	n transformation} We use the example and
o	n treats the right-hand side as the most
a	n tree for each 16 kbyte input block, enc
a	n tree in the output stream, then encodes
i	n turn, set $L[i]$ to be the
i	n turn, set $R[i]$ to the
o	n unusual data. Like the algorithm of Man
a	n use a single set of probabilities table
e	n using the positions of the suffixes in
i	n value at a given point in the vector $R
e	n we present modifications that improve t
e	n when the block size is quite large. Ho
i	n which codes that have not been seen in
i	n with ch appear in the {\em same order
i	n with ch. In our exam
o	n with Huffman or arithmetic coding. Bri
o	n with figures given by Bell~\cite{bell}.

Figure 1: Example of sorted rotations. Twenty consecutive rotations from the sorted list of rotations of a version of this paper are shown, together with the final character of each rotation.

Burrows M, Wheeler DJ: A block sorting lossless data compression algorithm. *Digital Equipment Corporation, Palo Alto, CA* 1994, Technical Report 124; 1994

FM Index: querying



Colors show what parts of matrix are shown on right

final char (L)
a
o
o
o
o
a
a
i
i
o
a
e
i
e
e
i
i
i
o
o

Figure 1: Example of sorted rotations. Twenty consecutive rotations from the sorted list of rotations of a version of this paper are shown, together with the final character of each rotation.

Burrows M, Wheeler DJ: A block sorting lossless data compression algorithm. *Digital Equipment Corporation, Palo Alto, CA* 1994, Technical Report 124; 1994

FM Index: querying

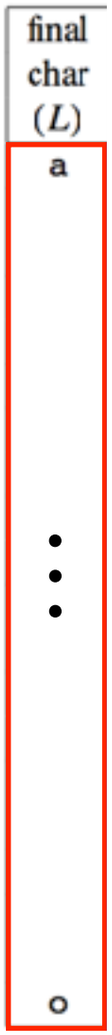
final char (L)
a
o
o
o
o
a
a
i
i
o
a
e
i
e
e
i
i
i
o
o

Figure 1: Example of sorted rotations. Twenty consecutive rotations from the sorted list of rotations of a version of this paper are shown, together with the final character of each rotation.

Burrows M, Wheeler DJ: A block sorting lossless data compression algorithm. *Digital Equipment Corporation, Palo Alto, CA* 1994, Technical Report 124; 1994

FM Index: querying

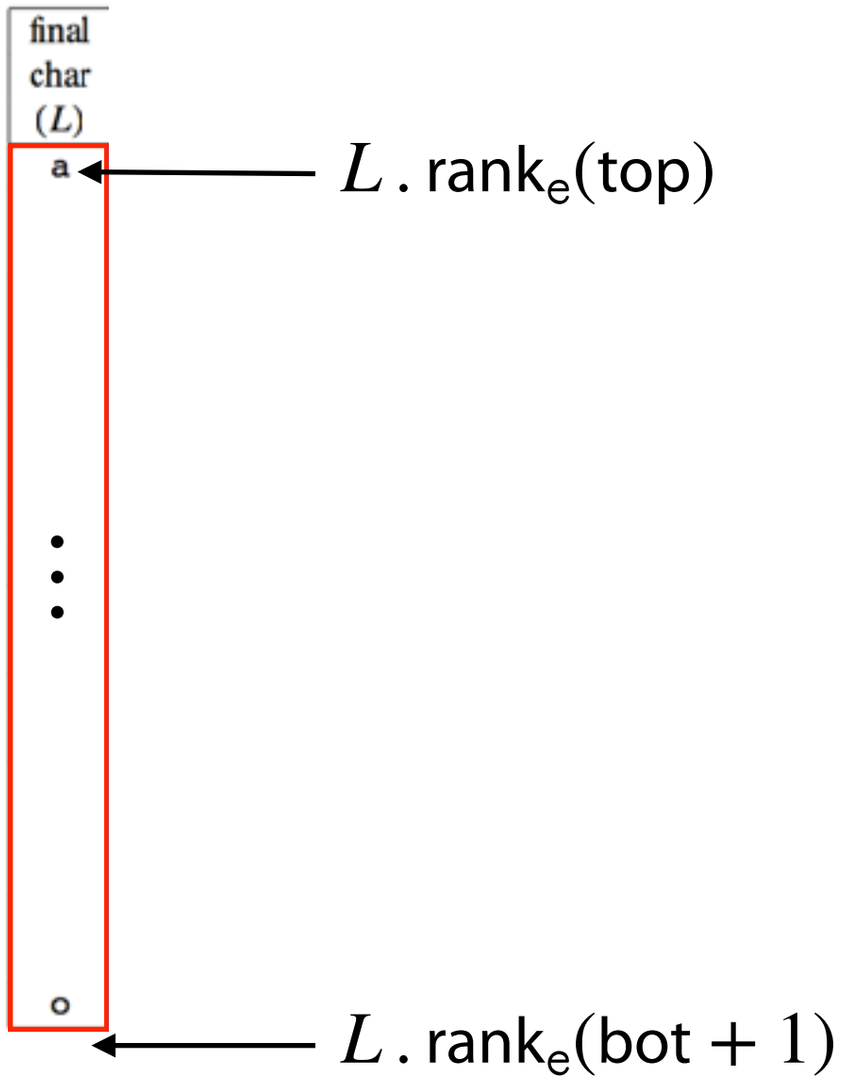
What e's are in this range?



Burrows M, Wheeler DJ: A block sorting lossless data compression algorithm. *Digital Equipment Corporation, Palo Alto, CA* 1994, Technical Report 124; 1994

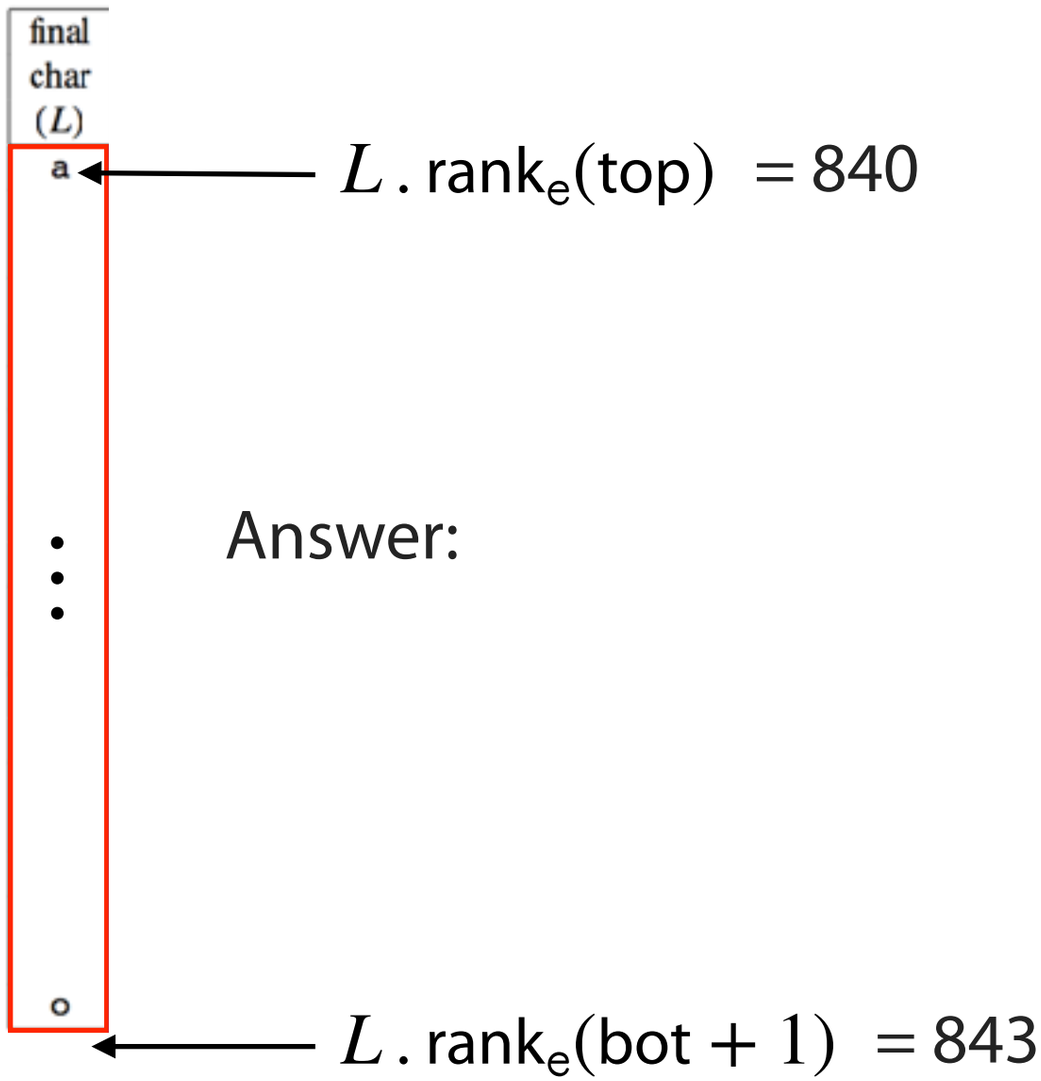
FM Index: querying

What e's are in this range?



Burrows M, Wheeler DJ: A block sorting lossless data compression algorithm. *Digital Equipment Corporation, Palo Alto, CA* 1994, Technical Report 124; 1994

FM Index: querying

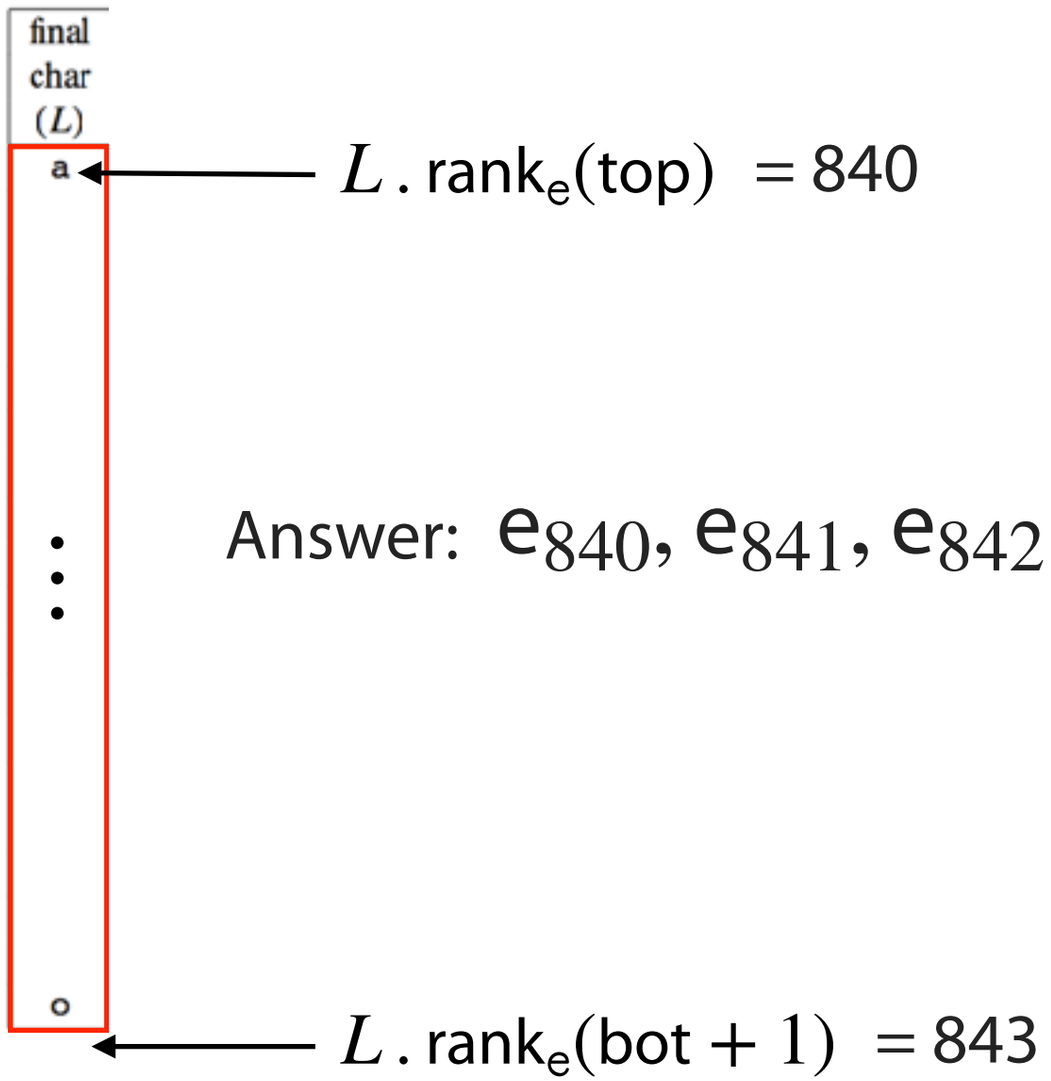


What e's are in this range?

Answer:

Burrows M, Wheeler DJ: A block sorting lossless data compression algorithm. *Digital Equipment Corporation, Palo Alto, CA* 1994, Technical Report 124; 1994

FM Index: querying



What e's are in this range?

Answer: $e_{840}, e_{841}, e_{842}$

Burrows M, Wheeler DJ: A block sorting lossless data compression algorithm. *Digital Equipment Corporation, Palo Alto, CA* 1994, Technical Report 124; 1994

FM Index: querying

$P = \mathbf{aba}$

F						L
$	a	b	a	a	b	**a_0**
a_0	$	a	b	a	a	**b_0**
a_1	a	b	a	$	a	**b_1**
a_2	b	a	$	a	b	**a_1**
a_3	b	a	a	b	a	**$**
b_0	a	$	a	b	a	**a_2**
b_1	a	a	b	a	$	**a_3**

Next char	Rank	Skip	Next range
a			

FM Index: querying

$P =$ **aba**

F						*L*
$	a	b	a	a	b	**a**$_0$
a$_0$	$	a	b	a	a	**b**$_0$
a$_1$	a	b	a	$	a	**b**$_1$
a$_2$	b	a	$	a	b	**a**$_1$
a$_3$	b	a	a	b	a	**$**
b$_0$	a	$	a	b	a	**a**$_2$
b$_1$	a	a	b	a	$	**a**$_3$

Next char	Rank	Skip	Next range
a		1 x $ = 1 1 x $ + 5 x **a** = 5	1 5

FM Index: querying

$P = \mathbf{aba}$

F						*L*
$	a	b	a	a	b	$\mathbf{a_0}$
$\mathbf{a_0}$	$	a	b	a	a	$\mathbf{b_0}$
$\mathbf{a_1}$	a	b	a	$	a	$\mathbf{b_1}$
$\mathbf{a_2}$	b	a	$	a	b	$\mathbf{a_1}$
$\mathbf{a_3}$	b	a	a	b	a	**$**
$\mathbf{b_0}$	a	$	a	b	a	$\mathbf{a_2}$
$\mathbf{b_1}$	a	a	b	a	$	$\mathbf{a_3}$

Next char	Rank	Skip	Next range
a		1 x $ = 1 1 x $ + 5 x **a** = 5	1 5

FM Index: querying

$P = \mathbf{aba}$

F						L
$	a	b	a	a	b	**a**$_0$
a$_0$	$	a	b	a	a	**b**$_0$
a$_1$	a	b	a	$	a	**b**$_1$
a$_2$	b	a	$	a	b	**a**$_1$
a$_3$	b	a	a	b	a	**$**
b$_0$	a	$	a	b	a	**a**$_2$
b$_1$	a	a	b	a	$	**a**$_3$

Next char	Rank	Skip	Next range
a		1 x $ = 1 1 x $ + 5 x **a** = 5	1 5
b			

FM Index: querying

$P = \mathbf{aba}$

F						L
$	a	b	a	a	b	**a**$_0$
a$_0$	$	a	b	a	a	**b**$_0$
a$_1$	a	b	a	$	a	**b**$_1$
a$_2$	b	a	$	a	b	**a**$_1$
a$_3$	b	a	a	b	a	**$**
b$_0$	a	$	a	b	a	**a**$_2$
b$_1$	a	a	b	a	$	**a**$_3$

Next char	Rank	Skip	Next range
a		1 x $ = 1 1 x $ + 5 x **a** = 5	1 5
b	$L.\text{rank}_{\text{b}}(1) = 0$ $L.\text{rank}_{\text{b}}(5) = 2$	1 x $ + 5 x **a** = 5	0 + 5 = 5 2 + 5 = 7

FM Index: querying

$P = \mathbf{aba}$

F						L
$	a	b	a	a	b	$\mathbf{a}_0$
$\mathbf{a}_0$	$	a	b	a	a	$\mathbf{b}_0$
$\mathbf{a}_1$	a	b	a	$	a	$\mathbf{b}_1$
$\mathbf{a}_2$	b	a	$	a	b	$\mathbf{a}_1$
$\mathbf{a}_3$	b	a	a	b	a	**$**
$\mathbf{b}_0$	a	$	a	b	a	$\mathbf{a}_2$
$\mathbf{b}_1$	a	a	b	a	$	$\mathbf{a}_3$

Next char	Rank	Skip	Next range
a		1 x $ = 1 1 x $ + 5 x **a** = 5	1 5
b	$L.\text{rank}_\text{b}(1) = 0$ $L.\text{rank}_\text{b}(5) = 2$	1 x $ + 5 x **a** = 5	0 + 5 = 5 2 + 5 = 7
a			

FM Index: querying

$P = \mathbf{aba}$

F						L
$	a	b	a	a	b	$\mathbf{a}_0$
$\mathbf{a}_0$	$	a	b	a	a	$\mathbf{b}_0$
$\mathbf{a}_1$	a	b	a	$	a	$\mathbf{b}_1$
$\mathbf{a}_2$	b	a	$	a	b	$\mathbf{a}_1$
$\mathbf{a}_3$	b	a	a	b	a	**$**
$\mathbf{b}_0$	a	$	a	b	a	$\mathbf{a}_2$
$\mathbf{b}_1$	a	a	b	a	$	$\mathbf{a}_3$

Next char	Rank	Skip	Next range
a		1 x $ = 1 1 x $ + 5 x **a** = 5	1 5
b	$L.\text{rank}_b(1) = 0$ $L.\text{rank}_b(5) = 2$	1 x $ + 5 x **a** = 5	0 + 5 = 5 2 + 5 = 7
a	$L.\text{rank}_a(5) = 2$ $L.\text{rank}_a(7) = 4$	1 x $ = 1	0 + 1 = 3 2 + 1 = 5

FM Index: querying

$P = \mathbf{aba}$

F						L
$	a	b	a	a	b	$\mathbf{a}_0$
$\mathbf{a}_0$	$	a	b	a	a	$\mathbf{b}_0$
$\mathbf{a}_1$	a	b	a	$	a	$\mathbf{b}_1$
$\mathbf{a}_2$	b	a	$	a	b	$\mathbf{a}_1$
$\mathbf{a}_3$	b	a	a	b	a	**$**
$\mathbf{b}_0$	a	$	a	b	a	$\mathbf{a}_2$
$\mathbf{b}_1$	a	a	b	a	$	$\mathbf{a}_3$

Next char	Rank	Skip	Next range
a		1 x $ = 1 1 x $ + 5 x **a** = 5	1 5
b	$L.\text{rank}_b(1) = 0$ $L.\text{rank}_b(5) = 2$	1 x $ + 5 x **a** = 5	0 + 5 = 5 2 + 5 = 7
a	$L.\text{rank}_a(5) = 2$ $L.\text{rank}_a(7) = 4$	1 x $ = 1	0 + 1 = 3 2 + 1 = 5

FM Index: querying

$P = \mathbf{aba}$

F						L
$	a	b	a	a	b	$\mathbf{a_0}$
$\mathbf{a_0}$	$	a	b	a	a	$\mathbf{b_0}$
$\mathbf{a_1}$	a	b	a	$	a	$\mathbf{b_1}$
$\mathbf{a_2}$	b	a	$	a	b	$\mathbf{a_1}$
$\mathbf{a_3}$	b	a	a	b	a	**$**
$\mathbf{b_0}$	a	$	a	b	a	$\mathbf{a_2}$
$\mathbf{b_1}$	a	a	b	a	$	$\mathbf{a_3}$

$T.\text{count}(\text{aba}) = 2$

Next char	Rank	Skip	Next range
a		1 x $ = 1 1 x $ + 5 x **a** = 5	1 5
b	$L.\text{rank}_b(1) = 0$ $L.\text{rank}_b(5) = 2$	1 x $ + 5 x **a** = 5	0 + 5 = 5 2 + 5 = 7
a	$L.\text{rank}_a(5) = 2$ $L.\text{rank}_a(7) = 4$	1 x $ = 1	0 + 1 = 3 2 + 1 = 5

FM Index: querying

FM index match(P): 

top $\leftarrow 0$

bot $\leftarrow |T|$

$i \leftarrow |P| - 1$

while $i \geq 0$ and bot $>$ top

  $c \leftarrow P[i]$

  top $\leftarrow$ BWT . $C[c]$ + BWT . rank$_c$(top)

  bot $\leftarrow$ BWT . $C[c]$ + BWT . rank$_c$(bot)

  $i \leftarrow i - 1$

return (top, bot)

FM Index: querying

FM index match(P): length-m string

O ( )

top $\leftarrow 0$

bot $\leftarrow |T|$

$i \leftarrow |P| - 1$

while $i \geq 0$ and bot $>$ top

Skip Rank

$c \leftarrow P[i]$

top $\leftarrow$ BWT . $C[c]$ + BWT . rank$_c$(top)

bot $\leftarrow$ BWT . $C[c]$ + BWT . rank$_c$(bot)

$i \leftarrow i - 1$

return (top, bot)

FM Index: querying

FM index match(P): length-m string

$$O\left(\ m\ \right)$$

steps

top $\leftarrow 0$

bot $\leftarrow |T|$

$i \leftarrow |P| - 1$

while $i \geq 0$ and bot $>$ top

Skip Rank

$c \leftarrow P[i]$

top $\leftarrow$ BWT . $C[c]$ + BWT . rank$_c$(top)

bot $\leftarrow$ BWT . $C[c]$ + BWT . rank$_c$(bot)

$i \leftarrow i - 1$

return (top, bot)

FM Index: querying

FM index match(P): length-m string

$$O\left(m \log_2 \sigma \right)$$

steps rank queries

top $\leftarrow 0$
bot $\leftarrow |T|$
$i \leftarrow |P| - 1$
while $i \geq 0$ and bot $>$ top
    $c \leftarrow P[i]$
    top $\leftarrow$ BWT . $C[c]$ + BWT . rank$_c$(top)
    bot $\leftarrow$ BWT . $C[c]$ + BWT . rank$_c$(bot)
    $i \leftarrow i - 1$
return (top, bot)

Skip Rank

FM Index: querying

Full-text index answers queries:

Locate(P), where $P \in \Sigma^m$, returns all offsets where P matches a substring of T

Count(P) returns # of offsets where P matches a substring of T

Extract(i, m) returns $T[i : i + m - 1]$ (length-m substring starting at i)

FM Index: querying

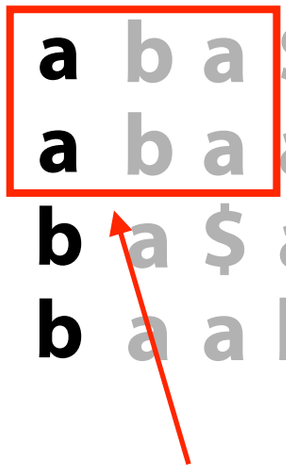
F						*L*
$	a	b	a	a	b	**a**
a	$	a	b	a	a	**b**
a	a	b	a	$	a	**b**
a	b	a	$	a	b	**a**
a	b	a	a	b	a	**$**
b	a	$	a	b	a	**a**
b	a	a	b	a	$	**a**

FM Index: querying

F						*L*
$	a	b	a	a	b	**a**
a	$	a	b	a	a	**b**
a	a	b	a	$	a	**b**
a	b	a	$	a	b	**a**
a	b	a	a	b	a	**$**
b	a	$	a	b	a	**a**
b	a	a	b	a	$	**a**

FM Index: querying

F						*L*
$	a	b	a	a	b	**a**
a	$	a	b	a	a	**b**
a	a	b	a	$	a	**b**
a	b	a	$	a	b	**a**
a	b	a	a	b	a	**$**
b	a	$	a	b	a	**a**
b	a	a	b	a	$	**a**



What offsets do these occur at in *T*?

FM Index: querying

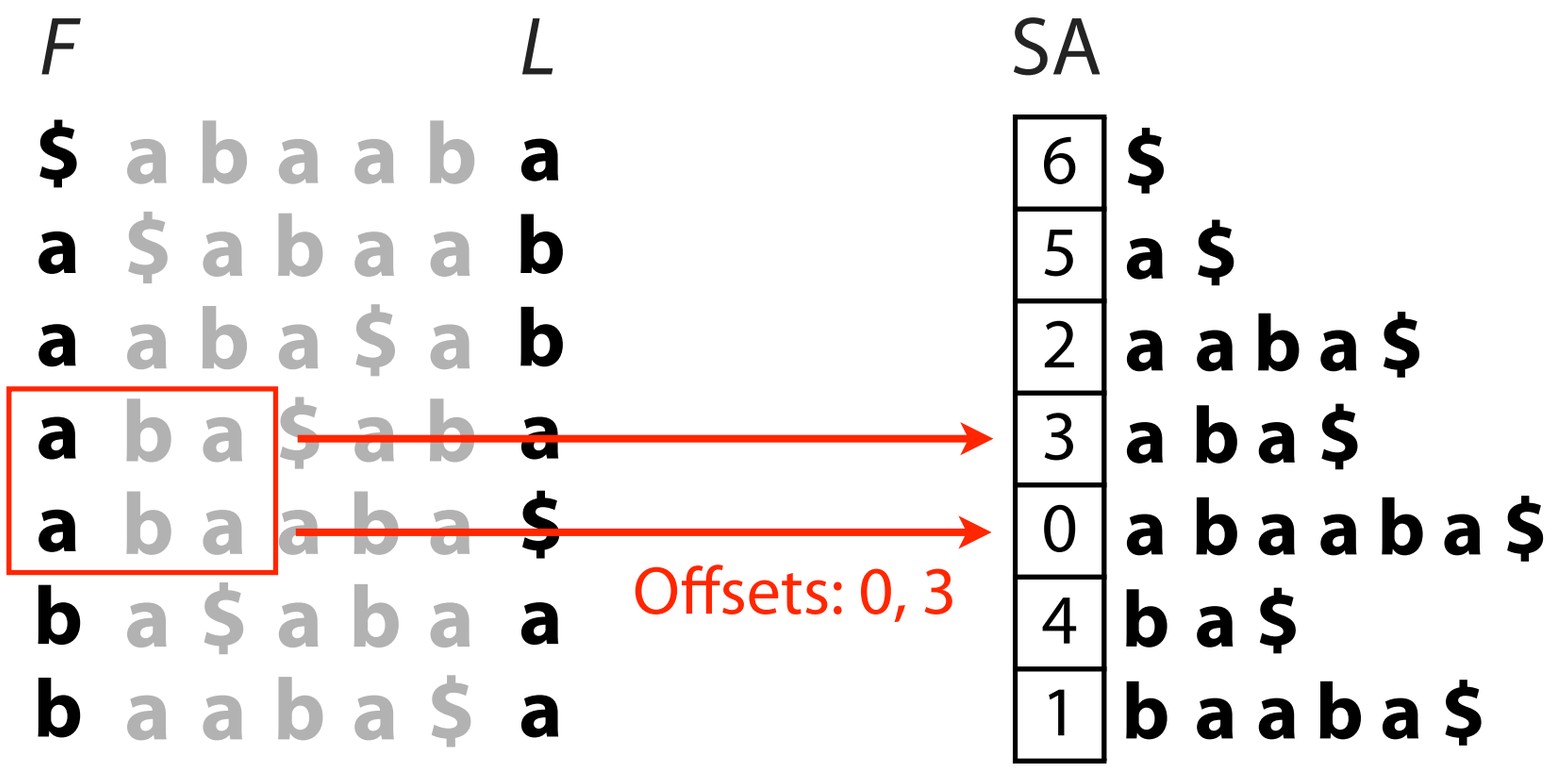
A precomputed "suffix array" stores the answers…

F	*L*
$ a b a a b	**a**
a $ a b a a	**b**
a a b a $ a	**b**
a b a $ a b	**a**
a b a a b a	**$**
b a $ a b a	**a**
b a a b a $	**a**

SA	
6	**$**
5	**a $**
2	**a a b a $**
3	**a b a $**
0	**a b a a b a $**
4	**b a $**
1	**b a a b a $**

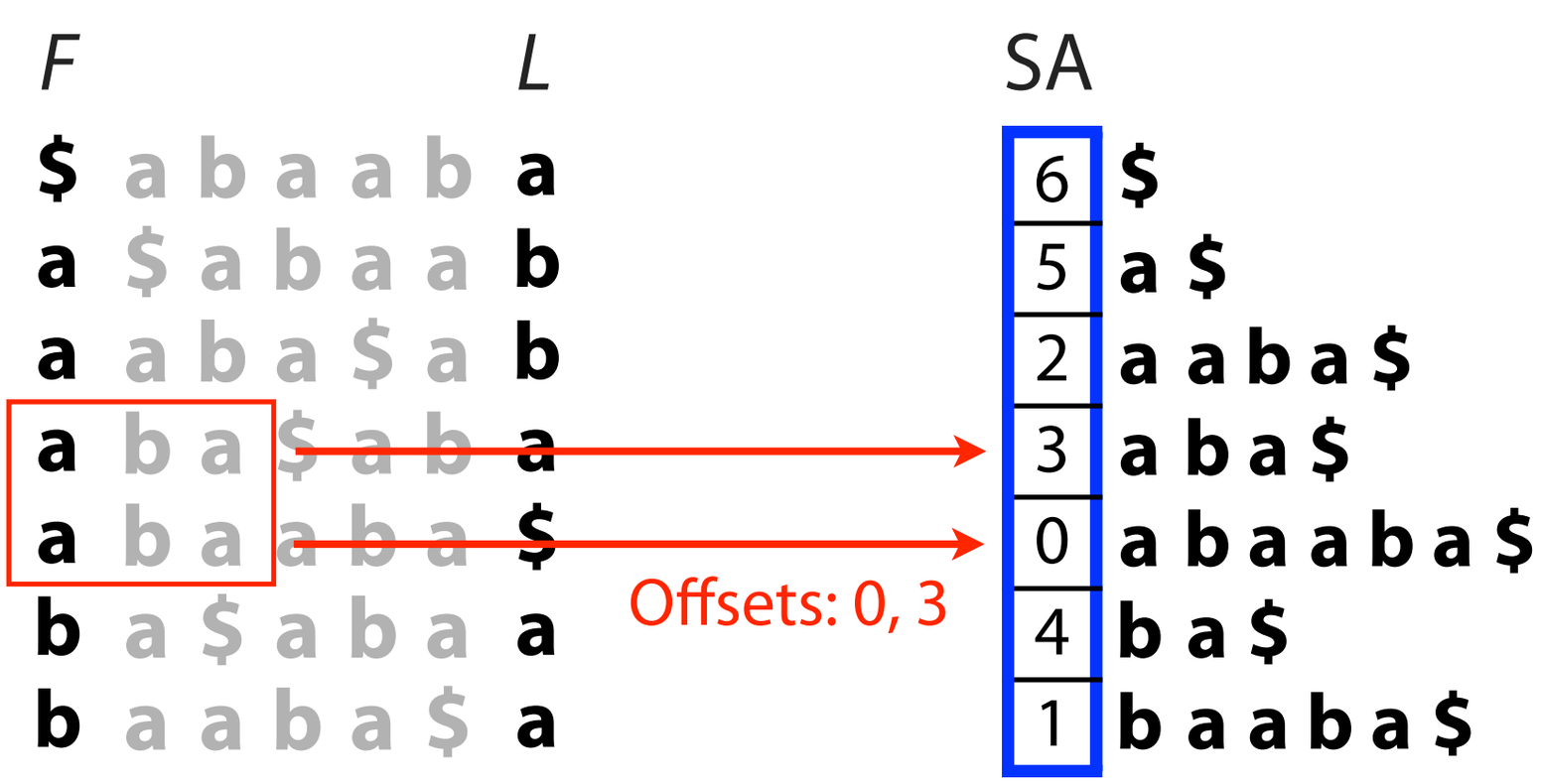
FM Index: querying

A precomputed "suffix array" stores the answers…



FM Index: querying

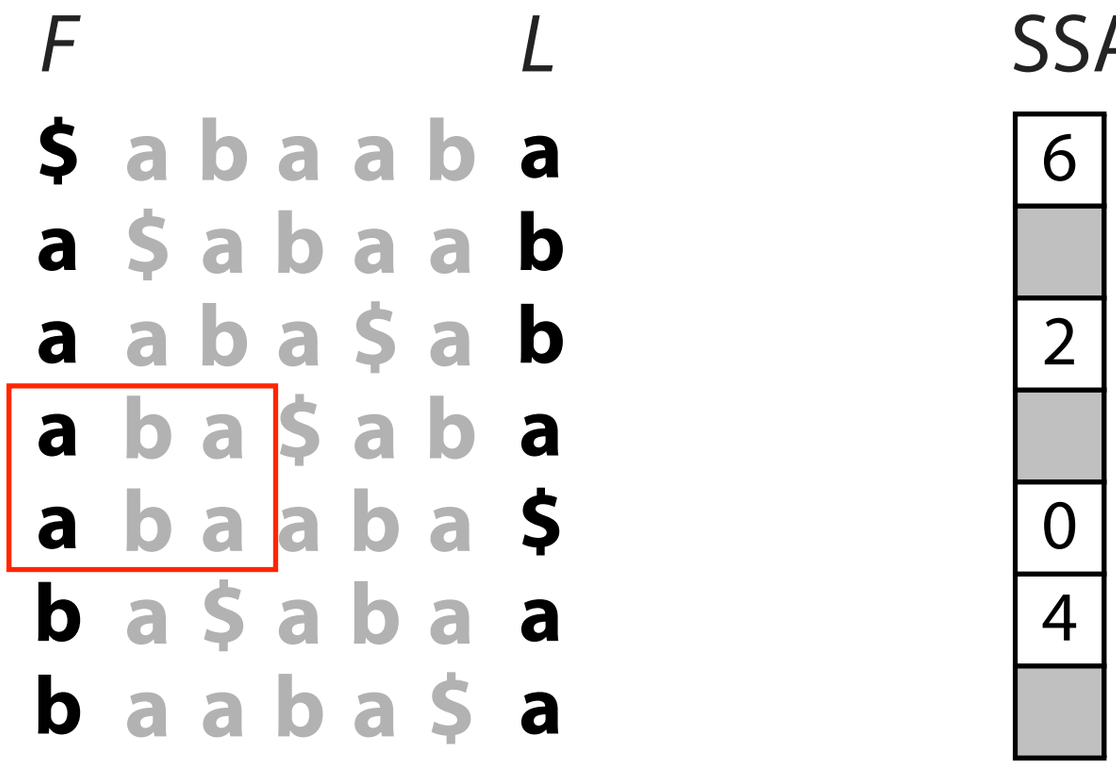
A precomputed "suffix array" stores the answers…



But n integers is too big!

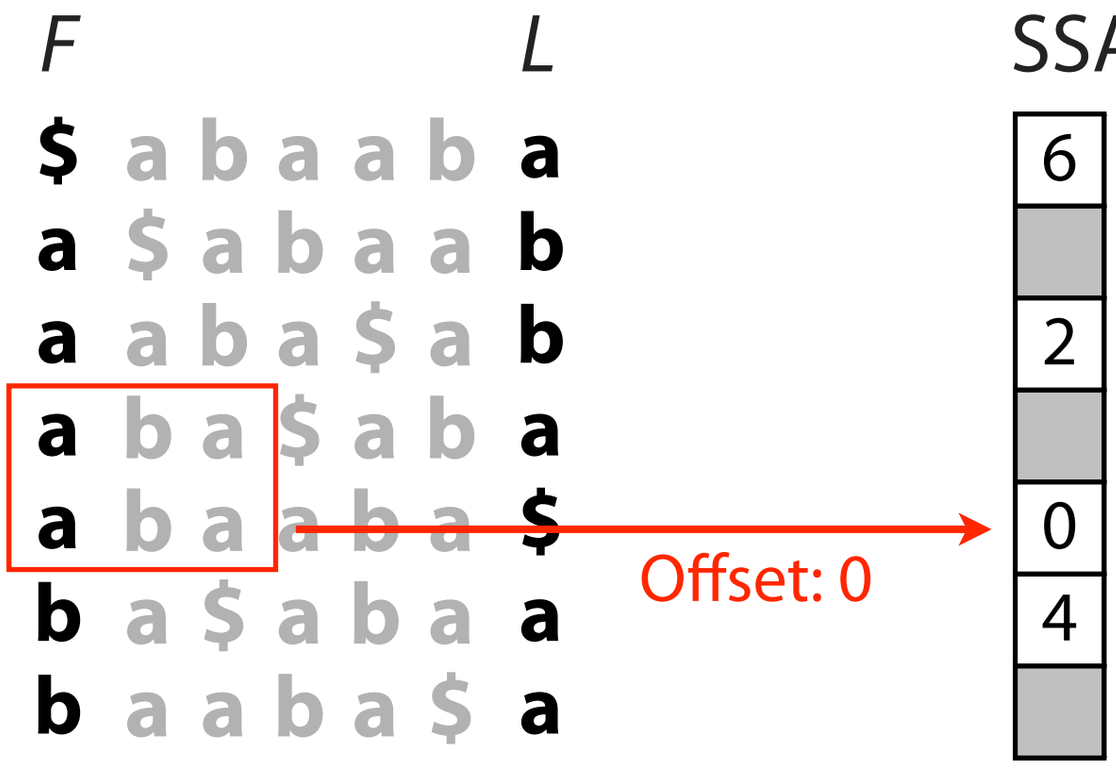
FM Index: querying

A **sampled** suffix array stores **some** answers…



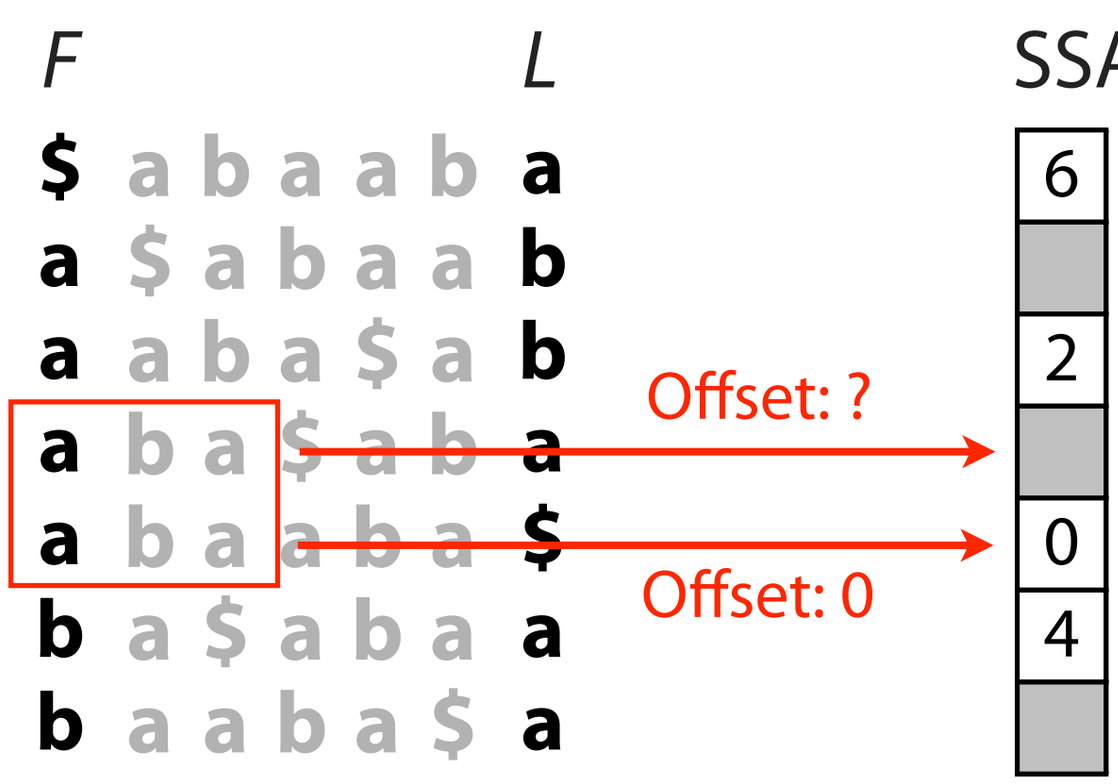
FM Index: querying

A **sampled** suffix array stores **some** answers…



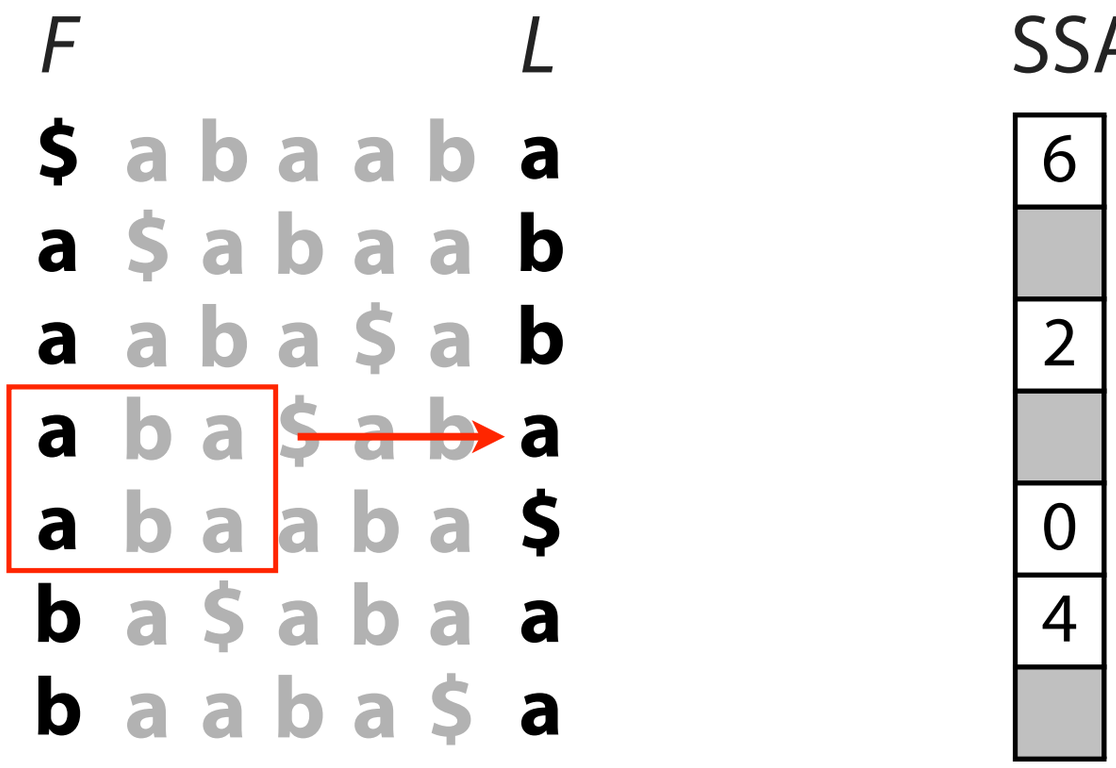
FM Index: querying

A **sampled** suffix array stores **some** answers…



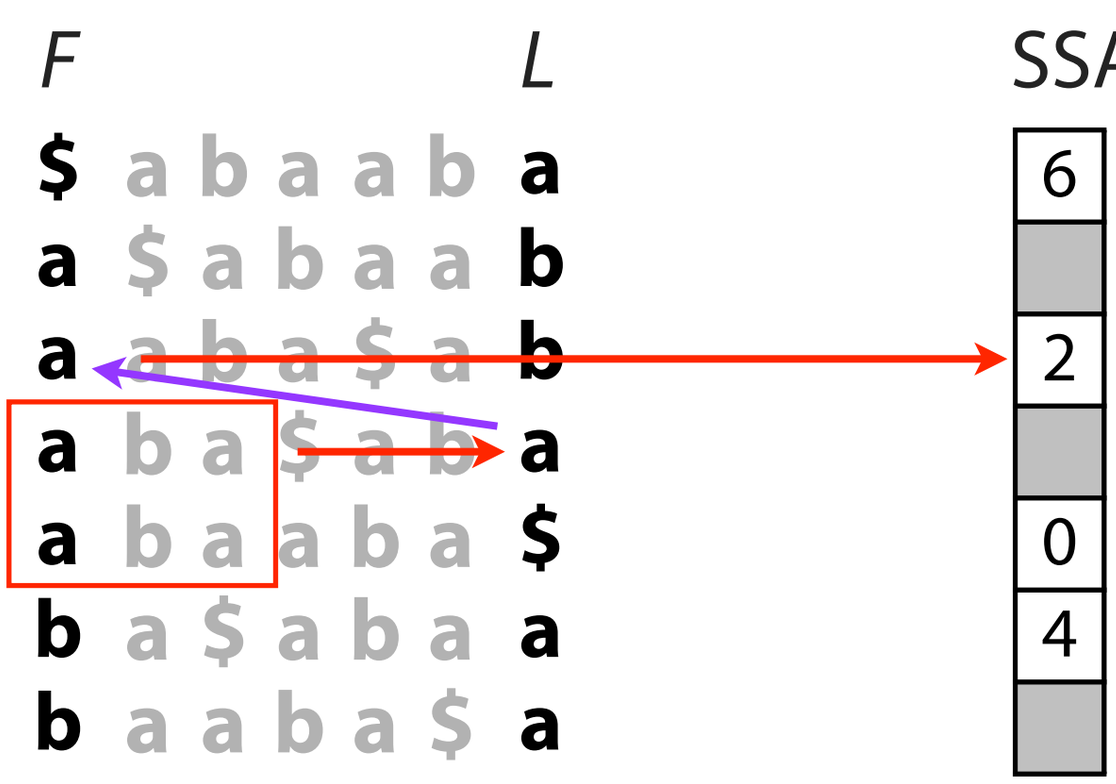
FM Index: querying

With LF mapping we "walk to the nearest" answer



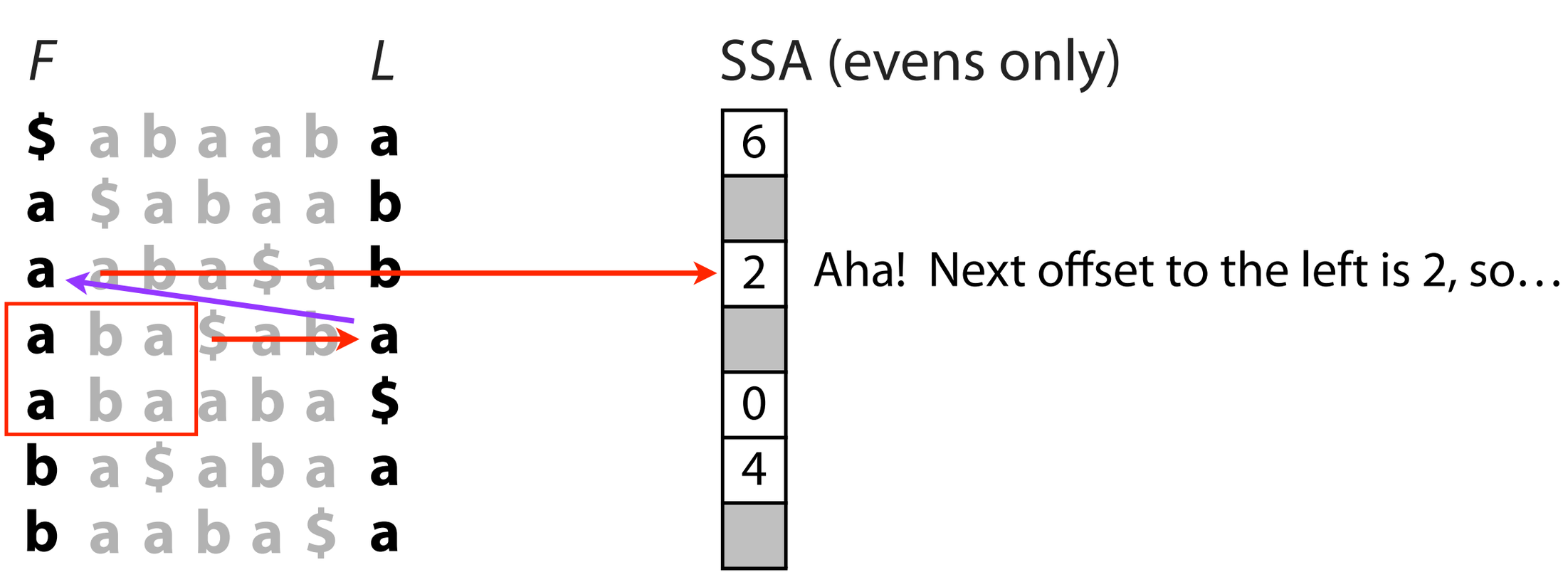
FM Index: querying

With LF mapping we "walk to the nearest" answer



FM Index: querying

With LF mapping we "walk to the nearest" answer



FM Index: querying

With LF mapping we "walk to the nearest" answer