FM Index: Efficient matching with BWT

Ben Langmead

Department of Computer Science

Please sign guestbook (www.langmead-lab.org/teaching-materials) to tell me briefly how you are using the slides. For original Keynote files, email me (ben.langmead@gmail.com).
Wavelet trees

Armed with Wavelet Trees, let’s return to the Burrows-Wheeler Transform

We can reverse it efficiently now!
Burrows-Wheeler Transform

\[ T: a_0 \ b_0 \ a_1 \ a_2 \ b_1 \ a_3 \$ \]

LF Mapping: The \(i^{th}\) occurrence of a character \(c\) in \(L\) and the \(i^{th}\) occurrence of \(c\) in \(F\) correspond to the \textit{same} occurrence in \(T\) (i.e. have same rank)
Burrows-Wheeler Transform

\[ T: \text{a}_0 \text{b}_0 \text{a}_1 \text{a}_2 \text{b}_1 \text{a}_3 \$ \]

LF Mapping: The \( i^{\text{th}} \) occurrence of a character \( c \) in \( L \) and the \( i^{\text{th}} \) occurrence of \( c \) in \( F \) correspond to the same occurrence in \( T \) (i.e. have same rank)
Burrows-Wheeler Transform

$F$

$L$

F

$\\$

a

a

a

a

a

b

b

a

a

a

b

a

b

a

aa

abba$a$aa
Burrows-Wheeler Transform

\[ \text{WT}(L) = \]

\[
\text{abba}\$aa
\]

\[\begin{array}{c}
0110100 \\
\end{array}\]

\[
\begin{array}{cc}
0 & 1 \\
\end{array}
\]

\[
\begin{array}{cc}
110 & \\
\end{array}
\]

\[
\begin{array}{cc}
\$ & b \\
\end{array}
\]
Recall: 1st row has $ in $F$, so start there

In $L$, we see an a. What is its rank?
Recall: 1st row has $ in $, so start there

In $L$, we see an a. What is its rank?

$$\text{WT}(L) \cdot \text{rank}_a(0) = 0$$
Burrows-Wheeler Transform

<table>
<thead>
<tr>
<th>$ F $</th>
<th>$ L $</th>
<th>Rank</th>
<th>Skip</th>
<th>Next row</th>
<th>Next char</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ $</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td>a</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Burrows-Wheeler Transform

<table>
<thead>
<tr>
<th>$F$</th>
<th>$L$</th>
<th>Rank</th>
<th>Skip</th>
<th>Next row</th>
<th>Next char</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rightarrow$</td>
<td>$\rightarrow$</td>
<td>$a$</td>
<td>$L \cdot \text{rank}_a(0) = 0$</td>
<td>$1 \times $ = 1</td>
<td>$0 + 1 = 1$</td>
</tr>
<tr>
<td>$a$</td>
<td>$b$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>$$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Burrows-Wheeler Transform

<table>
<thead>
<tr>
<th>$F$</th>
<th>$L$</th>
<th>Rank</th>
<th>Skip</th>
<th>Next row</th>
<th>Next char</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rightarrow $</td>
<td>$\rightarrow \text{a}$</td>
<td>$L. \ rank_a(0) = 0$</td>
<td>$1 \times $ = 1$</td>
<td>$0 + 1 = 1$</td>
<td>$\text{a}$</td>
</tr>
<tr>
<td>$\text{a}$</td>
<td>$\rightarrow \text{b}$</td>
<td></td>
<td>$1 \times $ + 4 \times \text{a} = 5$</td>
<td>$0 + 5 = 5$</td>
<td>$\text{b}$</td>
</tr>
<tr>
<td>$\text{a}$</td>
<td>$\text{b}$</td>
<td></td>
<td></td>
<td></td>
<td>$\text{a}$</td>
</tr>
<tr>
<td>$\text{a}$</td>
<td>$\text{a}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{a}$</td>
<td>$$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{b}$</td>
<td>$\rightarrow \text{a}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{b}$</td>
<td>$\text{a}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Burrows-Wheeler Transform

<table>
<thead>
<tr>
<th>$F$</th>
<th>$L$</th>
<th>Rank</th>
<th>Skip</th>
<th>Next row</th>
<th>Next char</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rightarrow $$</td>
<td>$a$</td>
<td>$L \cdot \text{rank}_a(0) = 0$</td>
<td>$1 \times $$ = 1</td>
<td>$0 + 1 = 1$</td>
<td>$a$</td>
</tr>
<tr>
<td>$a$</td>
<td>$b$</td>
<td>$L \cdot \text{rank}_b(1) = 0$</td>
<td>$1 \times $$ + 4 \times a = 5$</td>
<td>$0 + 5 = 5$</td>
<td>$b$</td>
</tr>
<tr>
<td>$a$</td>
<td>$b$</td>
<td>$L \cdot \text{rank}_a(5) = 2$</td>
<td>$1 \times $$ = 1$</td>
<td>$2 + 1 = 3$</td>
<td>$a$</td>
</tr>
<tr>
<td>$a$</td>
<td>$a$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$a$</td>
<td>$$$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$b$</td>
<td>$a$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$b$</td>
<td>$a$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Skip amount can be looked up; pre-calculate $C$ where $C[c]$ (c is a character) equals the number of characters alphabetically smaller than $c$ in $T$
Burrows-Wheeler Transform

<table>
<thead>
<tr>
<th>$F$</th>
<th>$L$</th>
<th>Rank</th>
<th>Skip</th>
<th>Next row</th>
<th>Next char</th>
</tr>
</thead>
<tbody>
<tr>
<td>→ $</td>
<td>$ a</td>
<td></td>
<td></td>
<td></td>
<td>a</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td></td>
<td>1 x $ = 1$</td>
<td>0 + 1 = 1</td>
<td>b</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td></td>
<td>1 x $ + 4 x a = 5$</td>
<td>0 + 5 = 5</td>
<td>a</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>a</td>
<td>L. rank$_a$(5) = 2</td>
<td>1 x $ = 1$</td>
<td>2 + 1 = 3</td>
<td>a</td>
</tr>
<tr>
<td>a</td>
<td>a</td>
<td>L. rank$_a$(3) = 1</td>
<td>1 x $ = 1$</td>
<td>1 + 1 = 2</td>
<td>b</td>
</tr>
<tr>
<td>a</td>
<td>$</td>
<td>$</td>
<td>L. rank$_b$(2) = 1</td>
<td>1 x $ + 4 x a = 5$</td>
<td>1 + 5 = 6</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>$</td>
<td>$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>L. rank$_a$(6) = 3</td>
<td>1 x $ = 1$</td>
<td>3 + 1 = 4</td>
<td>$</td>
</tr>
</tbody>
</table>

Skip amount can be looked up; pre-calculate $C$ where $C[c]$ ($c$ is a character) equals the number of characters alphabetically smaller than $c$ in $T$

Here, $C[$$] = 0$, $C[a] = 1$, $C[b] = 5$
## Burrows-Wheeler Transform

<table>
<thead>
<tr>
<th></th>
<th>Rank</th>
<th>Skip</th>
<th>Next row</th>
<th>Next char</th>
</tr>
</thead>
<tbody>
<tr>
<td>L. rank\textsubscript{a}(0) = 0</td>
<td>1 x $ = 1</td>
<td>0 + 1 = 1</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>L. rank\textsubscript{b}(1) = 0</td>
<td>1 x $ + 4 x a = 5</td>
<td>0 + 5 = 5</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>L. rank\textsubscript{a}(5) = 2</td>
<td>1 x $ = 1</td>
<td>2 + 1 = 3</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>L. rank\textsubscript{a}(3) = 1</td>
<td>1 x $ = 1</td>
<td>1 + 1 = 2</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>L. rank\textsubscript{b}(2) = 1</td>
<td>1 x $ + 4 x a = 5</td>
<td>1 + 5 = 6</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>L. rank\textsubscript{a}(6) = 3</td>
<td>1 x $ = 1</td>
<td>3 + 1 = 4</td>
<td>$</td>
<td></td>
</tr>
</tbody>
</table>

Reversing is $O\left( n \log_2 \sigma \right)$

Rank + skip = LF mapping
Principles of navigation

Use $WT(BWT(T))$ to reverse: $BWT(T) \rightarrow T$

How do we do indexing?
Indexing

When we add some auxiliary data structures to make it easier to answer indexing queries

Opportunistic Data Structures with Applications

Paolo Ferragina*  Giovanni Manzini†
Università di Pisa  Università del Piemonte Orientale

"FM Index"

Abstract

In this paper we address the issue of compressing and indexing data. We devise a data structure whose space occupancy is a function of the entropy of the underlying data set. We call the data structure opportunistic since its space occupancy is decreased when the input is compressible and this space reduction is achieved at no significant slowdown in the query performance. More precisely, its space occupancy is optimal in an information-content sense because a text $T[1,u]$ is stored using $O(H_k(T)) + o(1)$ bits per input symbol in the worst case, where $H_k(T)$ is the $k$th order empirical entropy of $T$ (the bound holds for any fixed $k$). Given an arbitrary string $P[1,p]$, the opportunistic data structure allows to search for the occurrences of $P$ in $T$ in $O(n + occlog occ)$ time (for any fixed $c > 0$). If data are

Indexing

A full-text index for text $T \in \Sigma^n$ is a structure giving efficient answers to 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 $i$)
A **full-text index** for text $T \in \Sigma^n$ is a structure giving efficient answers to queries:

- **Locate($P$)**, where $P \in \Sigma^m$, returns all offsets where $P$ matches a substring of $T$.
- **Count($P$)** returns the number 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

How to **find**, **count** and **locate** substrings matching a query?

$abaababa$

$abaababa$

$abaababa$

$abaababa$

$abaababa$

$abaababa$

$abaababa$
Observation 1: Rows with \textit{same prefix} are consecutive

Observation 2: Characters in \textit{last column} are those \textit{preceding} the prefixes (to their \textit{left} in T)
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 $a$

$[C[a], C[b]) = [1, 5)$

Subscripts are ranks in $L$
**FM Index: querying**

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

\[ P = aba \]

<table>
<thead>
<tr>
<th>$F$</th>
<th>$L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$$  a  b  a  a  b  $  a_0</td>
<td></td>
</tr>
<tr>
<td>$a_0$  $$  a  b  a  a  $  b_0</td>
<td></td>
</tr>
<tr>
<td>$a_1$  a  b  a  $  a  $  b_1</td>
<td></td>
</tr>
<tr>
<td>$a_2$  b  a  $  a  b  $  a_1</td>
<td></td>
</tr>
<tr>
<td>$a_3$  b  a  a  b  a  $  $</td>
<td></td>
</tr>
<tr>
<td>$b_0$  a  $  a  b  a  $  a_2</td>
<td></td>
</tr>
<tr>
<td>$b_1$  a  a  b  a  $  a_3</td>
<td></td>
</tr>
</tbody>
</table>

\[ P = aba \]

<table>
<thead>
<tr>
<th>$F$</th>
<th>$L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$$  a  b  a  a  b  $  a_0</td>
<td></td>
</tr>
<tr>
<td>$a_0$  $$  a  b  a  a  $  b_0</td>
<td></td>
</tr>
<tr>
<td>$a_1$  a  b  a  $  a  $  b_1</td>
<td></td>
</tr>
<tr>
<td>$a_2$  b  a  $  a  b  $  a_1</td>
<td></td>
</tr>
<tr>
<td>$a_3$  b  a  a  b  a  $  $</td>
<td></td>
</tr>
<tr>
<td>$b_0$  a  $  a  b  a  $  a_2</td>
<td></td>
</tr>
<tr>
<td>$b_1$  a  a  b  a  $  a_3</td>
<td></td>
</tr>
</tbody>
</table>

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

Now we have the rows with prefix $ba$
FM Index: querying

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

$$P = \text{aba}$$

$\begin{array}{l}
F \\
$ 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 \\
\end{array}$

Use LF Mapping

$\begin{array}{l}
F \\
$ 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 \\
\end{array}$

\[ \text{Now we have the rows with prefix } \text{aba} \]

\[ T \cdot \text{count(aba)} = 2 \]
FM Index: querying

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

$$P = \text{bba}$$

<table>
<thead>
<tr>
<th>F</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{a b a a b a}_0$</td>
<td></td>
</tr>
<tr>
<td>$\text{a}_0 \text{ a b a a b}_0$</td>
<td></td>
</tr>
<tr>
<td>$\text{a}_1 \text{ a b a } \text{a b}_1$</td>
<td></td>
</tr>
<tr>
<td>$\text{a}_2 \text{ b a } \text{a b}_1$</td>
<td></td>
</tr>
<tr>
<td>$\text{a}_3 \text{ b a a b a } \text{a}_2$</td>
<td></td>
</tr>
</tbody>
</table>

Rows with $\text{ba}$ prefix

No bs!
### FM Index: querying

**Query:** \( P = \text{aba} \)

<table>
<thead>
<tr>
<th>( F )</th>
<th>( L )</th>
<th>Next char</th>
<th>Rank</th>
<th>Skip</th>
<th>Next range</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>a b a a b</td>
<td>a</td>
<td>a</td>
<td>1 x $ = 1</td>
<td>1</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>b</td>
<td>5</td>
</tr>
<tr>
<td>a</td>
<td>a</td>
<td>b</td>
<td>a</td>
<td>b</td>
<td>1</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>a</td>
<td>$</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>1 x $ + 5 x a = 5</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>a</td>
<td>b</td>
<td>a</td>
<td>$</td>
<td>a_3</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>a</td>
<td>b</td>
<td>a</td>
<td>a_2</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>a</td>
<td>b</td>
<td>a</td>
<td>a_3</td>
</tr>
</tbody>
</table>
FM Index: querying

\[
P = \text{aba}
\]

<table>
<thead>
<tr>
<th>( F )</th>
<th>( L )</th>
<th>Next char</th>
<th>Rank</th>
<th>Skip</th>
<th>Next range</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ \text{aba} \text{a}_0$</td>
<td>$\text{a}_0$</td>
<td>a</td>
<td>a</td>
<td>1 x $ = 1$</td>
<td>1</td>
</tr>
<tr>
<td>$\text{a}_0 \text{aba} \text{a}_1$</td>
<td>$\text{a}_1$</td>
<td>a</td>
<td>b</td>
<td>1 x $ + 5 \times \text{a} = 5$</td>
<td>5</td>
</tr>
<tr>
<td>$\text{a}_1 \text{aba} \text{a}_2$</td>
<td>$\text{a}_2$</td>
<td>b</td>
<td>a</td>
<td>$L \cdot \text{rank}_b(1) = 0$</td>
<td>0 + 5 = 5</td>
</tr>
<tr>
<td>$\text{a}_3 \text{aba} \text{a}_3$</td>
<td>$\text{a}_3$</td>
<td>b</td>
<td>$L \cdot \text{rank}_b(5) = 2$</td>
<td>1 x $ + 5 \times \text{a} = 5$</td>
<td>2 + 5 = 7</td>
</tr>
</tbody>
</table>
### FM Index: querying

**Pattern:** \( P = \text{aba} \)

<table>
<thead>
<tr>
<th>( F )</th>
<th>( L )</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>\text{aba} a_0</td>
</tr>
<tr>
<td>\text{a}_0</td>
<td>$ \text{a} \text{b} \text{a} \text{a} \text{a} \text{b}_0</td>
</tr>
<tr>
<td>\text{a}_1</td>
<td>\text{a} \text{b} \text{a} \text{a} $ \text{a} \text{b}_1</td>
</tr>
<tr>
<td>\text{a}_2</td>
<td>\text{b} \text{a} $ \text{a} \text{b} \text{a} \text{a}_1</td>
</tr>
<tr>
<td>\text{a}_3</td>
<td>\text{b} \text{a} \text{a} \text{b} \text{a} $</td>
</tr>
<tr>
<td>$ \text{b}_0</td>
<td>\text{a} $ \text{a} \text{b} \text{a} \text{a}_2</td>
</tr>
<tr>
<td>\text{b}_1</td>
<td>\text{a} \text{a} \text{b} \text{a} $ \text{a}_3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Next char</th>
<th>Rank</th>
<th>Skip</th>
<th>Next range</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{a}</td>
<td>( L \cdot \text{rank}_b(1) = 0 )</td>
<td>( 1 \times $ = 1 ) ( 1 \times $ + 5 \times \text{a} = 5 )</td>
<td>1</td>
</tr>
<tr>
<td>\text{b}</td>
<td>( L \cdot \text{rank}_b(5) = 2 )</td>
<td></td>
<td>0 + 5 = 5 2 + 5 = 7</td>
</tr>
<tr>
<td>\text{a}</td>
<td>( L \cdot \text{rank}_a(5) = 2 ) ( L \cdot \text{rank}_a(7) = 4 )</td>
<td>( 1 \times $ = 1 )</td>
<td>0 + 1 = 3 2 + 1 = 5</td>
</tr>
</tbody>
</table>
**FM Index: querying**

\[ P = \text{aba} \]

<table>
<thead>
<tr>
<th>\hspace{1cm}</th>
<th>\hspace{1cm}</th>
<th>\hspace{1cm}</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F )</td>
<td>( L )</td>
<td></td>
</tr>
<tr>
<td>$ \ a \ b \ a \ a \ b \ a_0 )</td>
<td>( a_0 \ a \ b \ a \ a \ b_0 )</td>
<td></td>
</tr>
<tr>
<td>( a_1 \ a \ b \ a \ $ \ a \ b_1 )</td>
<td>( \text{a} )</td>
<td></td>
</tr>
<tr>
<td>( a_2 \ b \ a \ $ \ a \ b \ a_1 )</td>
<td>( \text{b} )</td>
<td></td>
</tr>
<tr>
<td>( a_3 \ b \ a \ a \ b \ a \ $ )</td>
<td>( \text{a} )</td>
<td></td>
</tr>
<tr>
<td>( b_0 \ a \ $ \ a \ b \ a \ a_2 )</td>
<td>( \text{b} )</td>
<td></td>
</tr>
<tr>
<td>( b_1 \ a \ a \ b \ a \ $ \ a_3 )</td>
<td>( \text{a} )</td>
<td></td>
</tr>
</tbody>
</table>

\[ T. \ count(aba) = 2 \]

<table>
<thead>
<tr>
<th>Next char</th>
<th>Rank</th>
<th>Skip</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{a} )</td>
<td>( \text{b} )</td>
<td>( \text{a} )</td>
</tr>
<tr>
<td>( \text{a} )</td>
<td>( L. \ rank_b(1) = 0 )</td>
<td>( 1 \times $ = 1 )</td>
</tr>
<tr>
<td>( \text{b} )</td>
<td>( L. \ rank_b(5) = 2 )</td>
<td>( 1 \times $ + 5 \times a = 5 )</td>
</tr>
<tr>
<td>( \text{a} )</td>
<td>( L. \ rank_a(5) = 2 )</td>
<td>( 1 \times $ = 1 )</td>
</tr>
<tr>
<td>( \text{a} )</td>
<td>( L. \ rank_a(7) = 4 )</td>
<td>( 2 + 1 = 5 )</td>
</tr>
</tbody>
</table>
FM Index: querying

FM index match($P$):

Given query string $P$

```
top ← 0
bot ← |T|
i ← |P| − 1
while $i ≥ 0$ and $bot > top$
    $c ← P[i]$
    $top ← BWT . C[c] + BWT . rank_c(top)$
    $bot ← BWT . C[c] + BWT . rank_c(bot)$
    $i ← i − 1$
return (top, bot)
```

(For simplicity, version starts with the all-inclusive range rather than using 2 initial $BWT . C[ . . . ]$ lookups to get the range for the length-1 suffix)
FM Index: querying

A **full-text index** for text \( T \in \Sigma^n \) is a structure giving efficient answers to 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

Where are these occurrences in $T$?
**FM Index: querying**

Where are these occurrences in $T$?

If we had suffix array, we could look up offsets...

<table>
<thead>
<tr>
<th>$F$</th>
<th>$L$</th>
<th>$SA$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a b a a b a$</td>
<td>$a b a a b$</td>
<td>6 $</td>
</tr>
<tr>
<td>$a b a a b$</td>
<td>$a b a$</td>
<td>5 $</td>
</tr>
<tr>
<td>$b a a b$</td>
<td>$a b a$</td>
<td>2 $</td>
</tr>
<tr>
<td>$a b a a b a$</td>
<td>$a b a$</td>
<td>3 $</td>
</tr>
<tr>
<td>$b a b a b$</td>
<td>$a b a$</td>
<td>0 $</td>
</tr>
<tr>
<td>$b a a b a a$</td>
<td>$a b$</td>
<td>4 $</td>
</tr>
<tr>
<td>$b a b a a$</td>
<td>$b a a b a$</td>
<td>1 $</td>
</tr>
</tbody>
</table>

Offsets: 0, 3
FM Index: resolving offsets

Sampled Suffix Array (SSA): store some suffix array elements, not all

SSA (evens only)

Lookup for row 4 succeeds

Lookup for row 3 fails - SA entry was discarded