Lorem Ipsum Dolor

Gene Compression & Pattern Matching

Run Length Encoding

- We replace a run of k consecutive occurrences of symbol s with only two symbols: k, followed by s.
- For example, run-length encoding would compress the string TTTTTGGGAAAACCCCCCA into 5T3G4A6C1A.
- Run-length encoding works well for strings having lots of long runs, but real genomes do not have many runs.
- It would therefore be nice if we could first manipulate the genome to convert repeats into runs and then apply run-length encoding to the resulting string.

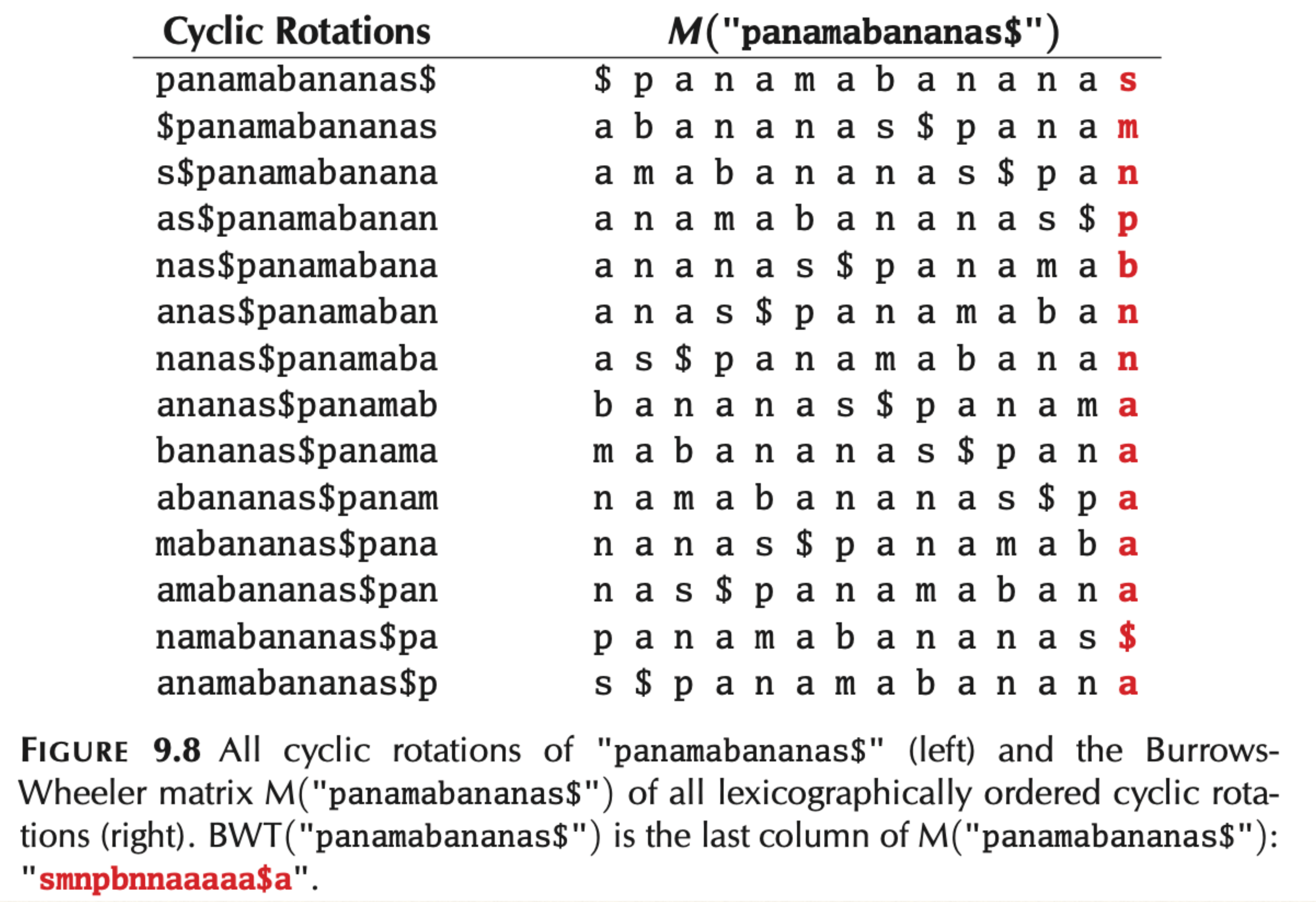
Run Length Encoding

- A naive way of creating runs in a string is to reorder the string's symbols lexico- graphically.
- For example, TACGTAACGATACGAT would become AAAAACCCGGGTTTT, which we could then compress into 5A3C3G4T.
- This method would represent a 3 GB human genome file using just four numbers.
- Ordering a string's symbols lexicographically is not suitable for compression because many different strings will get compressed into the same string.
- For example, the DNA strings GCATCATGCAT and ACTGACTACTG — as well as any string with the same nucleotide counts — get reordered into AAACCCGGTTT. As a result, we cannot decompress the compressed string, i.e., invert the compression operation to produce the original string.

Burrows-Wheeler transform

- First, form all possible cyclic rotations of Text; a cyclic rotation is defined by chopping off a suffix from the end of Text and appending this suffix to the beginning of Text.
    - Oorder all the cyclic rotations of Text lexicographically to form a |Text| x |Text| matrix of symbols that we call the Burrows-Wheeler matrix and denote by M(Text)

Burrows-Wheeler Matrix



Cyclic Rotations	*M*("panamabananas$")
panamabananas$	$panamabanana**s**
$panamabananas	abananas$pana**m**
s$panamabanana	amabananas$pa**n**
as$panamabanan	anamabananas$**p**
nas$panamabana	ananas$panama**b**
anas$panamaban	anas$panamaba**n**
nanas$panamaba	as$panamabana**n**
ananas$panamab	bananas$panam**a**
bananas$panama	mabananas$pan**a**
abananas$panam	namabananas$p**a**
mabananas$pana	nanas$panamab**a**
amabananas$pan	nas$panamaban**a**
namabananas$pa	panamabananas**$**
anamabananas$p	s$panamabanan**a**

Figure 9.8 All cyclic rotations of "panamabananas$" (left) and the Burrows-Wheeler matrix M("panamabananas$") of all lexicographically ordered cyclic rotations (right). BWT("panamabananas$") is the last column of M("panamabananas$"): "**smnpbnnaaaaa$a**".

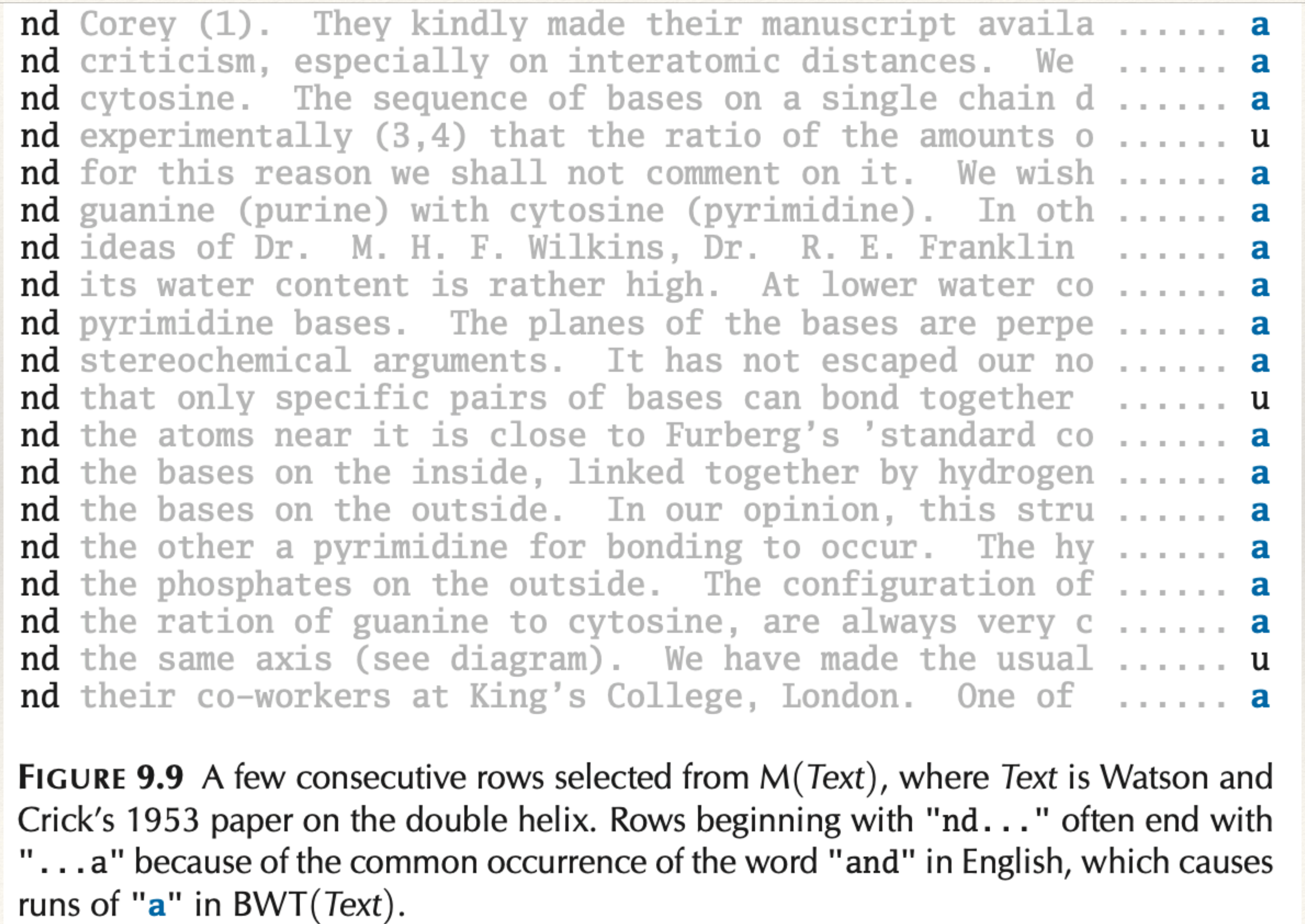
Burrows-Wheeler transform

- Notice that the first column of M(Text) contains the symbols of Text ordered lexico- graphically, which is just the naive rearrangement of Text
- Any column of M(Text) is some rearrange- ment of the symbols of Text.
- Last column of M(Text), called the Burrows-Wheeler transform of Text, or BWT(Text),
-

Repeat to runs

- If we re-examine the Burrows-Wheeler transform in Figure 9.8, we immediately notice that it has created the run "aaaaa" in BWT("panamabananas$") = "smnpbnnaaaaa$a".
- Suppose we perform BWT with some English text, now consider common words like “and”, “the”, … etc. In some rotation if they begin with “nd” they will end with “a”. Thus in the last column they will definitely create a run.
- These will be true for genomes where we have common words called repeats.

Francis and Creek



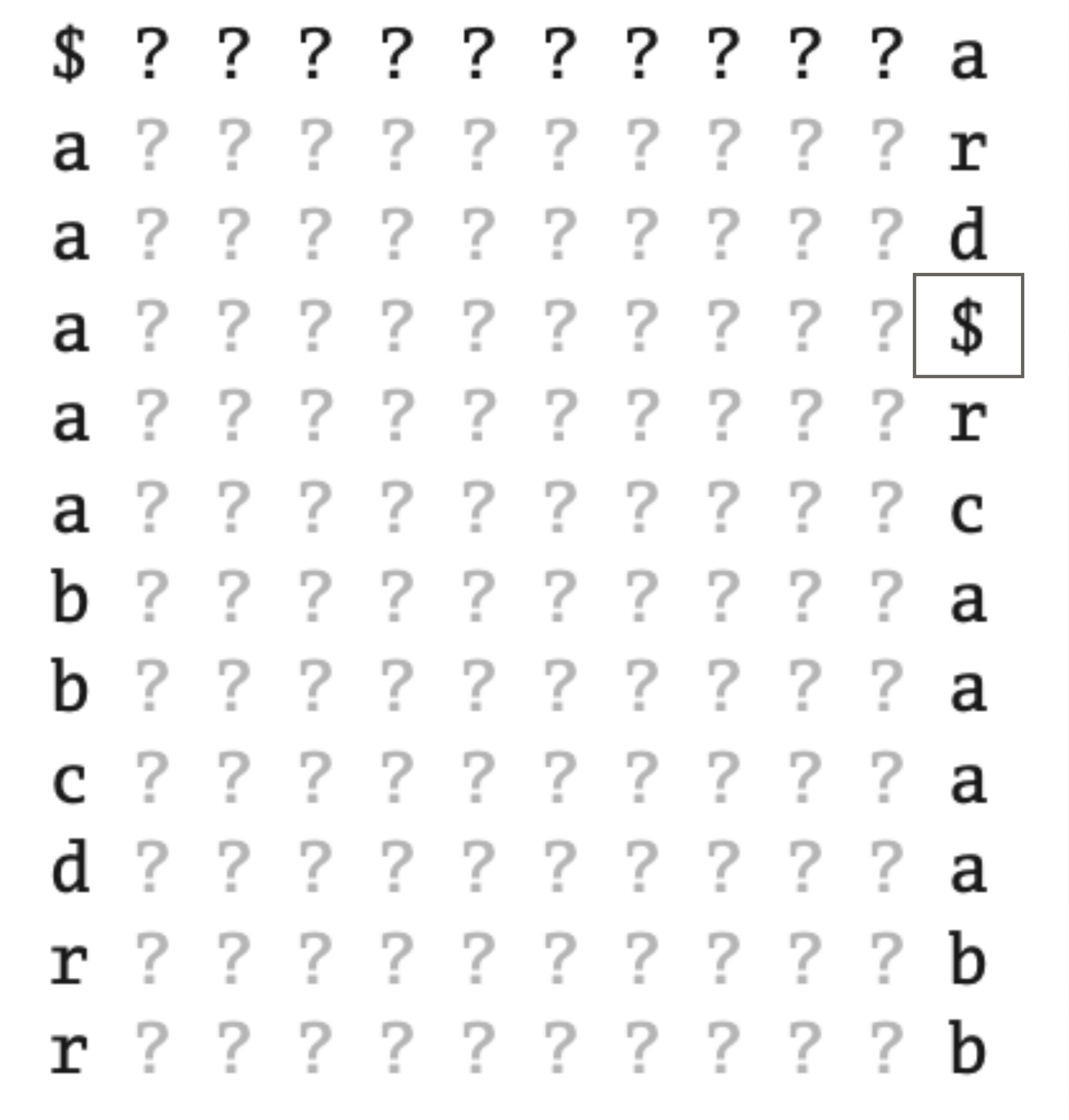
```
nd Corey (1).  They kindly made their manuscript availa ...... a
nd criticism, especially on interatomic distances.  We  ...... a
nd cytosine.  The sequence of bases on a single chain d ...... a
nd experimentally (3,4) that the ratio of the amounts o ...... u
nd for this reason we shall not comment on it.  We wish ...... a
nd guanine (purine) with cytosine (pyrimidine).  In oth ...... a
nd ideas of Dr.  M. H. F. Wilkins, Dr.  R. E. Franklin  ...... a
nd its water content is rather high.  At lower water co ...... a
nd pyrimidine bases.  The planes of the bases are perpe ...... a
nd stereochemical arguments.  It has not escaped our no ...... a
nd that only specific pairs of bases can bond together  ...... u
nd the atoms near it is close to Furberg’s ’standard co ...... a
nd the bases on the inside, linked together by hydrogen ...... a
nd the bases on the outside.  In our opinion, this stru ...... a
nd the other a pyrimidine for bonding to occur.  The hy ...... a
nd the phosphates on the outside.  The configuration of ...... a
nd the ration of guanine to cytosine, are always very c ...... a
nd the same axis (see diagram).  We have made the usual ...... u
nd their co-workers at King’s College, London.  One of  ...... a
```

FIGURE 9.9 A few consecutive rows selected from M(*Text*), where *Text* is Watson and Crick’s 1953 paper on the double helix. Rows beginning with "nd..." often end with "...a" because of the common occurrence of the word "and" in English, which causes runs of "a" in BWT(*Text*).

Inverting

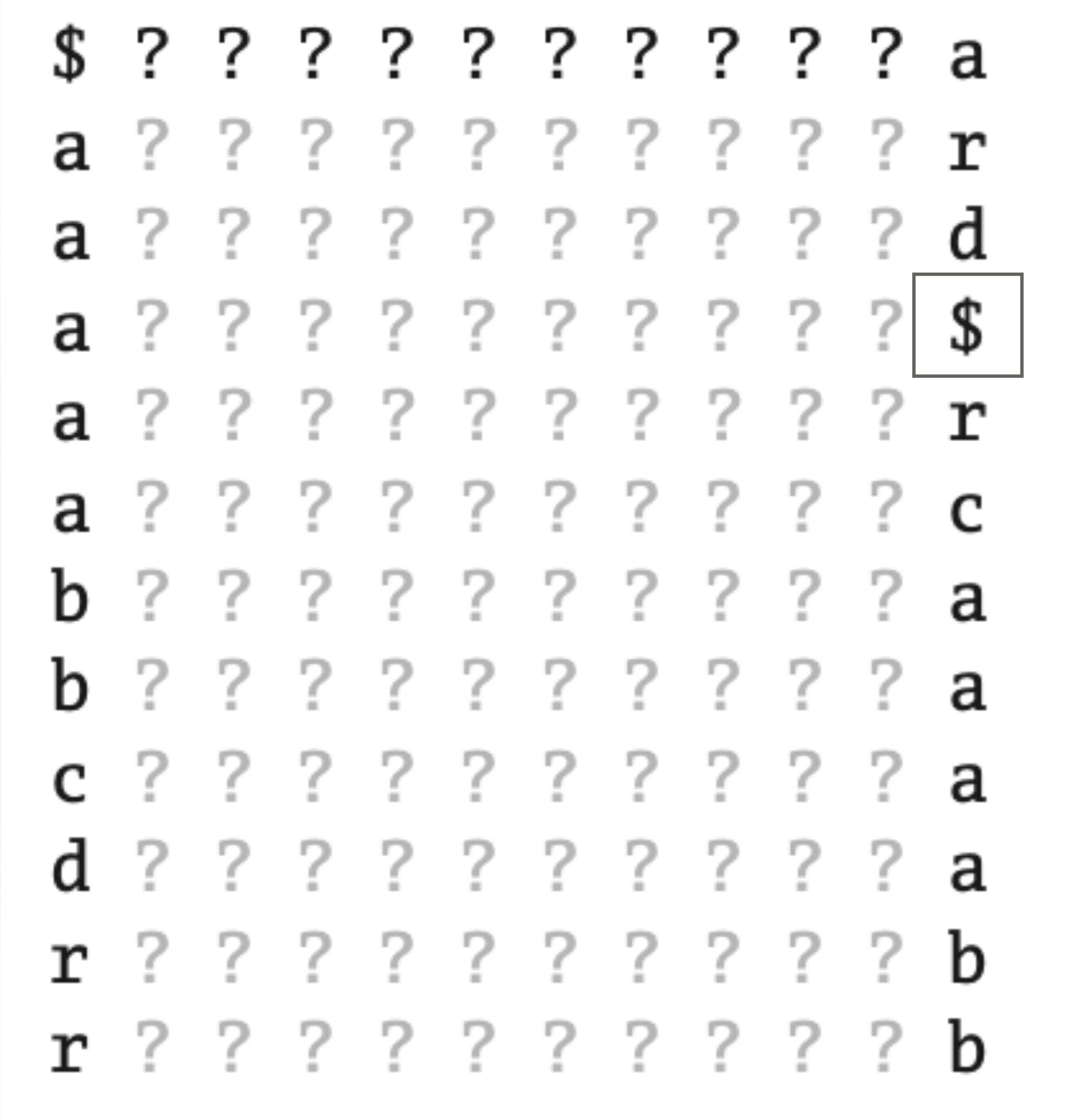
BWT(Text) = "ard$rcaaaabb".

Can you guess the first symbol?

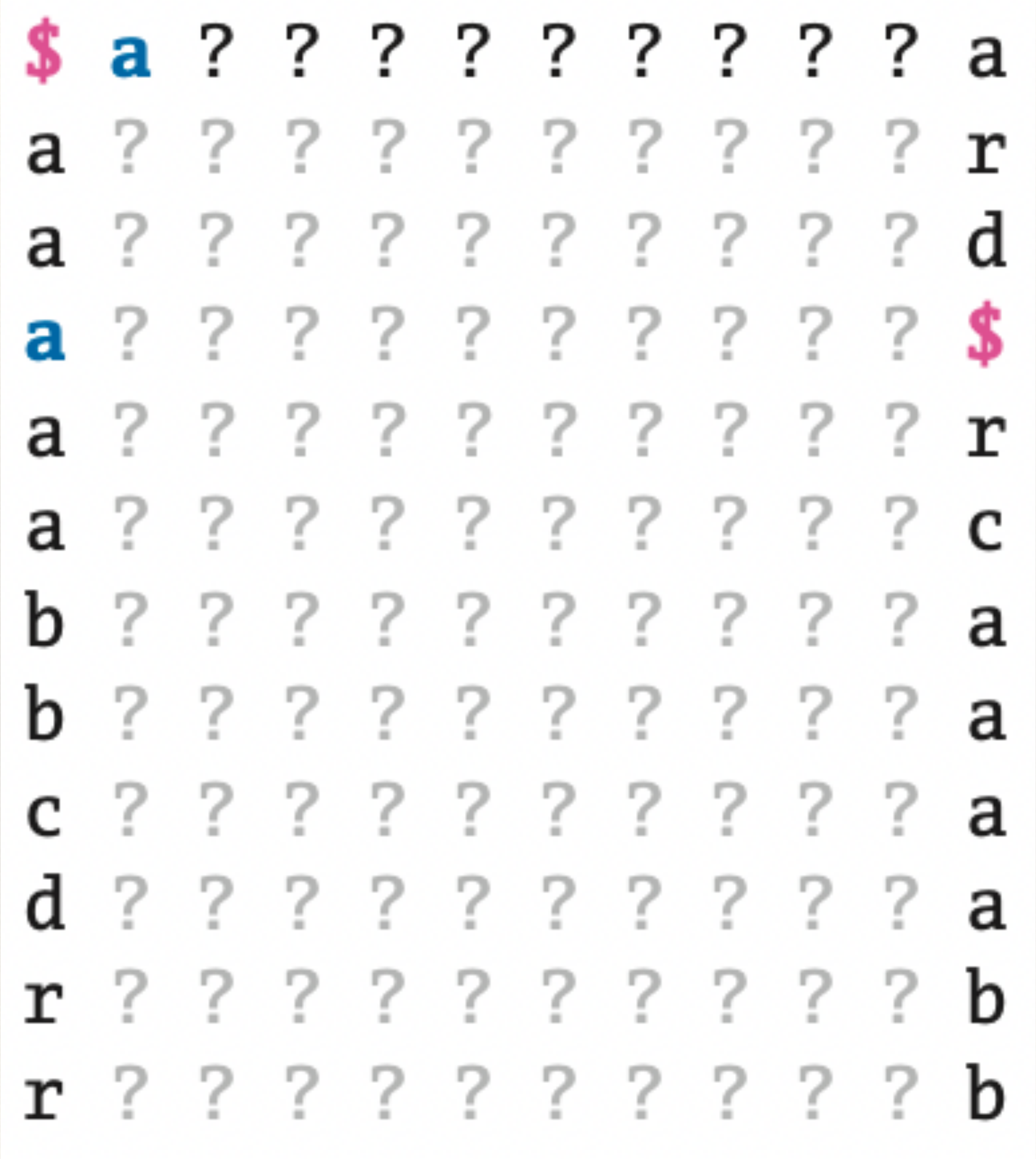


$	?	?	?	?	?	?	?	?	?	?	a
a	?	?	?	?	?	?	?	?	?	?	r
a	?	?	?	?	?	?	?	?	?	?	d
a	?	?	?	?	?	?	?	?	?	?	$
a	?	?	?	?	?	?	?	?	?	?	r
a	?	?	?	?	?	?	?	?	?	?	c
b	?	?	?	?	?	?	?	?	?	?	a
b	?	?	?	?	?	?	?	?	?	?	a
c	?	?	?	?	?	?	?	?	?	?	a
d	?	?	?	?	?	?	?	?	?	?	a
r	?	?	?	?	?	?	?	?	?	?	b
r	?	?	?	?	?	?	?	?	?	?	b

Inverting

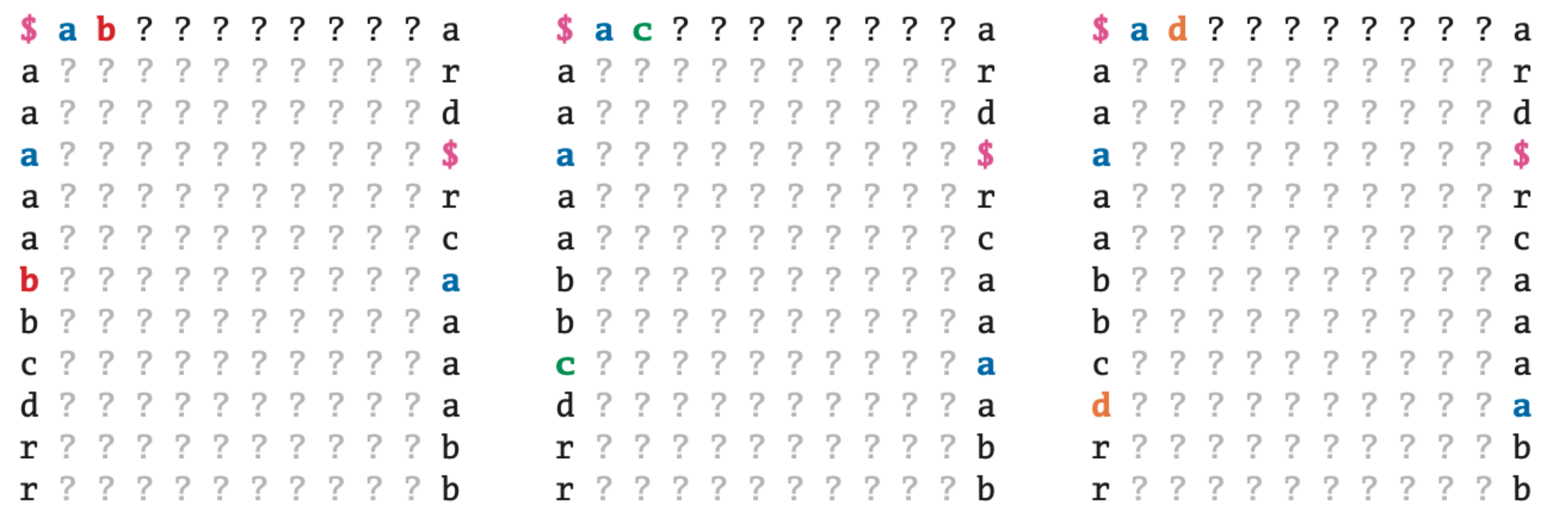


$	?	?	?	?	?	?	?	?	?	?	a
a	?	?	?	?	?	?	?	?	?	?	r
a	?	?	?	?	?	?	?	?	?	?	d
a	?	?	?	?	?	?	?	?	?	?	$
a	?	?	?	?	?	?	?	?	?	?	r
a	?	?	?	?	?	?	?	?	?	?	c
b	?	?	?	?	?	?	?	?	?	?	a
b	?	?	?	?	?	?	?	?	?	?	a
c	?	?	?	?	?	?	?	?	?	?	a
d	?	?	?	?	?	?	?	?	?	?	a
r	?	?	?	?	?	?	?	?	?	?	b
r	?	?	?	?	?	?	?	?	?	?	b



$	a	?	?	?	?	?	?	?	?	?	a
a	?	?	?	?	?	?	?	?	?	?	r
a	?	?	?	?	?	?	?	?	?	?	d
a	?	?	?	?	?	?	?	?	?	?	$
a	?	?	?	?	?	?	?	?	?	?	r
a	?	?	?	?	?	?	?	?	?	?	c
b	?	?	?	?	?	?	?	?	?	?	a
b	?	?	?	?	?	?	?	?	?	?	a
c	?	?	?	?	?	?	?	?	?	?	a
d	?	?	?	?	?	?	?	?	?	?	a
r	?	?	?	?	?	?	?	?	?	?	b
r	?	?	?	?	?	?	?	?	?	?	b

Second Symbol?

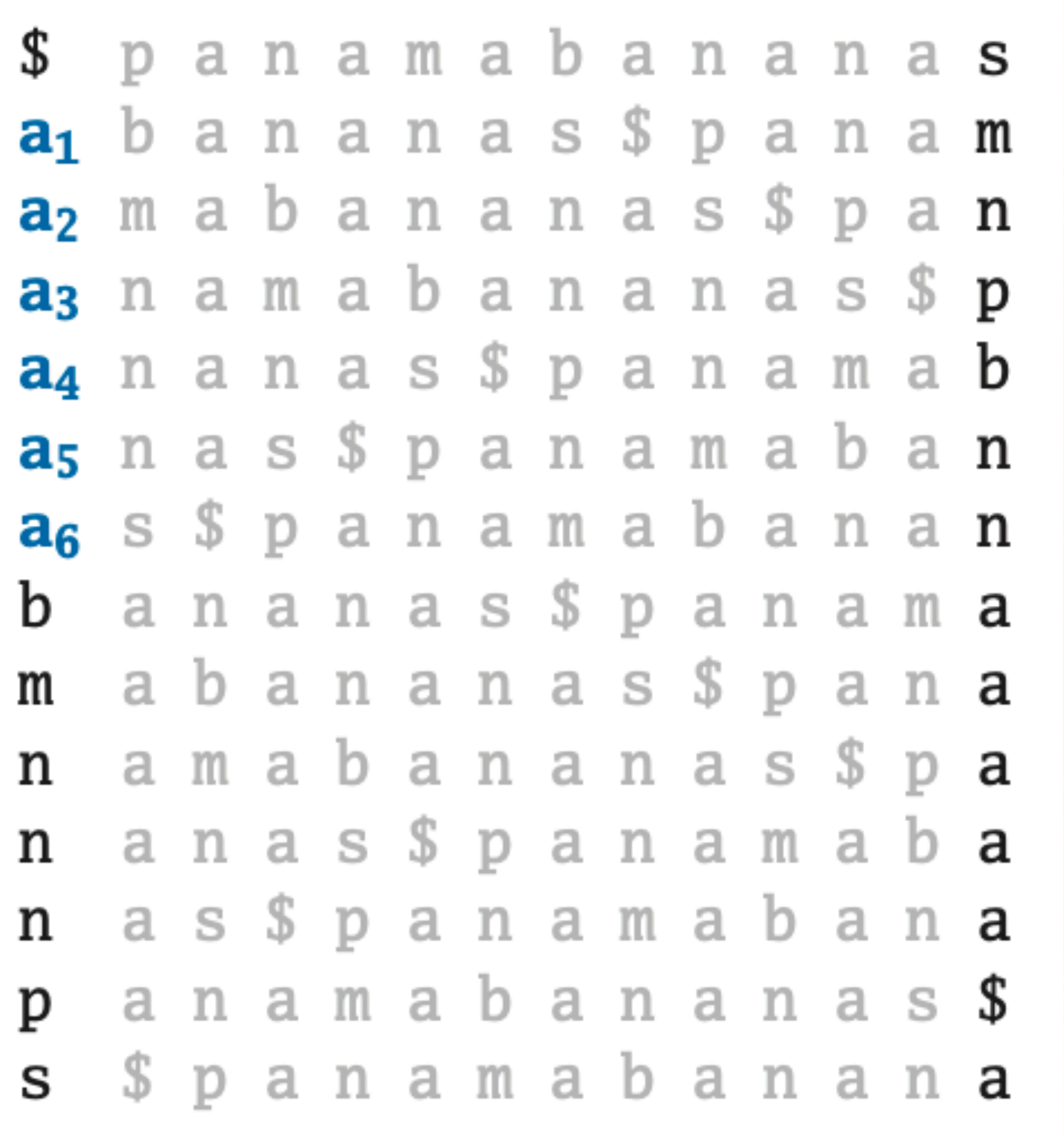


```
$ a b ? ? ? ? ? ? ? ? a     $ a c ? ? ? ? ? ? ? ? a     $ a d ? ? ? ? ? ? ? ? a
a ? ? ? ? ? ? ? ? ? ? r     a ? ? ? ? ? ? ? ? ? ? r     a ? ? ? ? ? ? ? ? ? ? r
a ? ? ? ? ? ? ? ? ? ? d     a ? ? ? ? ? ? ? ? ? ? d     a ? ? ? ? ? ? ? ? ? ? d
a ? ? ? ? ? ? ? ? ? ? $     a ? ? ? ? ? ? ? ? ? ? $     a ? ? ? ? ? ? ? ? ? ? $
a ? ? ? ? ? ? ? ? ? ? r     a ? ? ? ? ? ? ? ? ? ? r     a ? ? ? ? ? ? ? ? ? ? r
a ? ? ? ? ? ? ? ? ? ? c     a ? ? ? ? ? ? ? ? ? ? c     a ? ? ? ? ? ? ? ? ? ? c
b ? ? ? ? ? ? ? ? ? ? a     b ? ? ? ? ? ? ? ? ? ? a     b ? ? ? ? ? ? ? ? ? ? a
b ? ? ? ? ? ? ? ? ? ? a     b ? ? ? ? ? ? ? ? ? ? a     b ? ? ? ? ? ? ? ? ? ? a
c ? ? ? ? ? ? ? ? ? ? a     c ? ? ? ? ? ? ? ? ? ? a     c ? ? ? ? ? ? ? ? ? ? a
d ? ? ? ? ? ? ? ? ? ? a     d ? ? ? ? ? ? ? ? ? ? a     d ? ? ? ? ? ? ? ? ? ? a
r ? ? ? ? ? ? ? ? ? ? b     r ? ? ? ? ? ? ? ? ? ? b     r ? ? ? ? ? ? ? ? ? ? b
r ? ? ? ? ? ? ? ? ? ? b     r ? ? ? ? ? ? ? ? ? ? b     r ? ? ? ? ? ? ? ? ? ? b
```

FIGURE 9.10 The three possibilities ("b", "c", or "d") for the third element of the first row of M(*Text*) when BWT(*Text*) is "ard$rcaaaabb". One of these possibilities must correspond to the second symbol of *Text*.

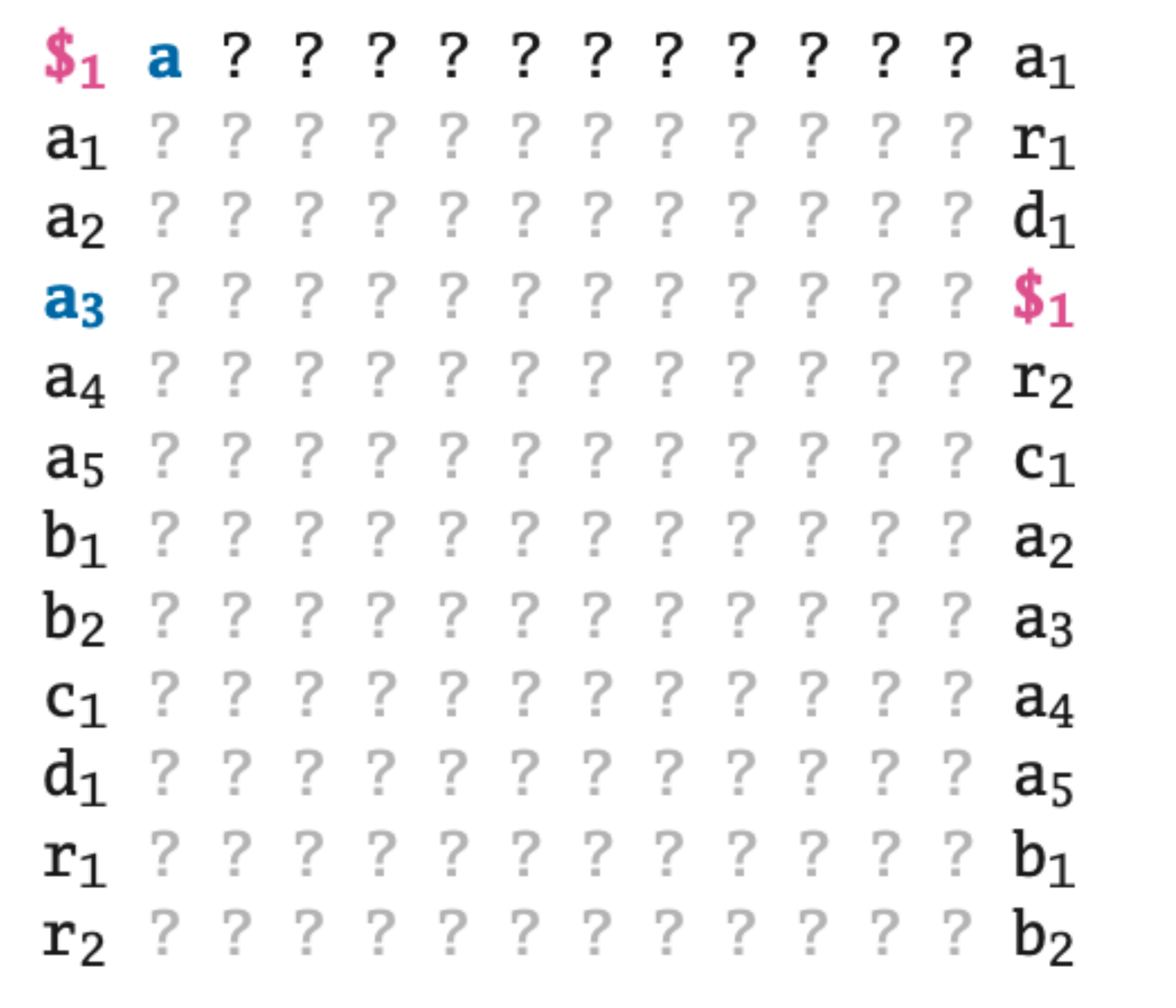
The First-Last Property

- Index the occurrences of each symbol in FirstColumn with subscripts according to their order of appearance in this column. When Text = "panamabananas$", six instances of "a" appear in FirstColumn.
- $pa_3na_2ma_1ba_4na_5na_6s\$$
- Six instances of "a" appear in exactly the same order in FirstColumn and LastColumn.

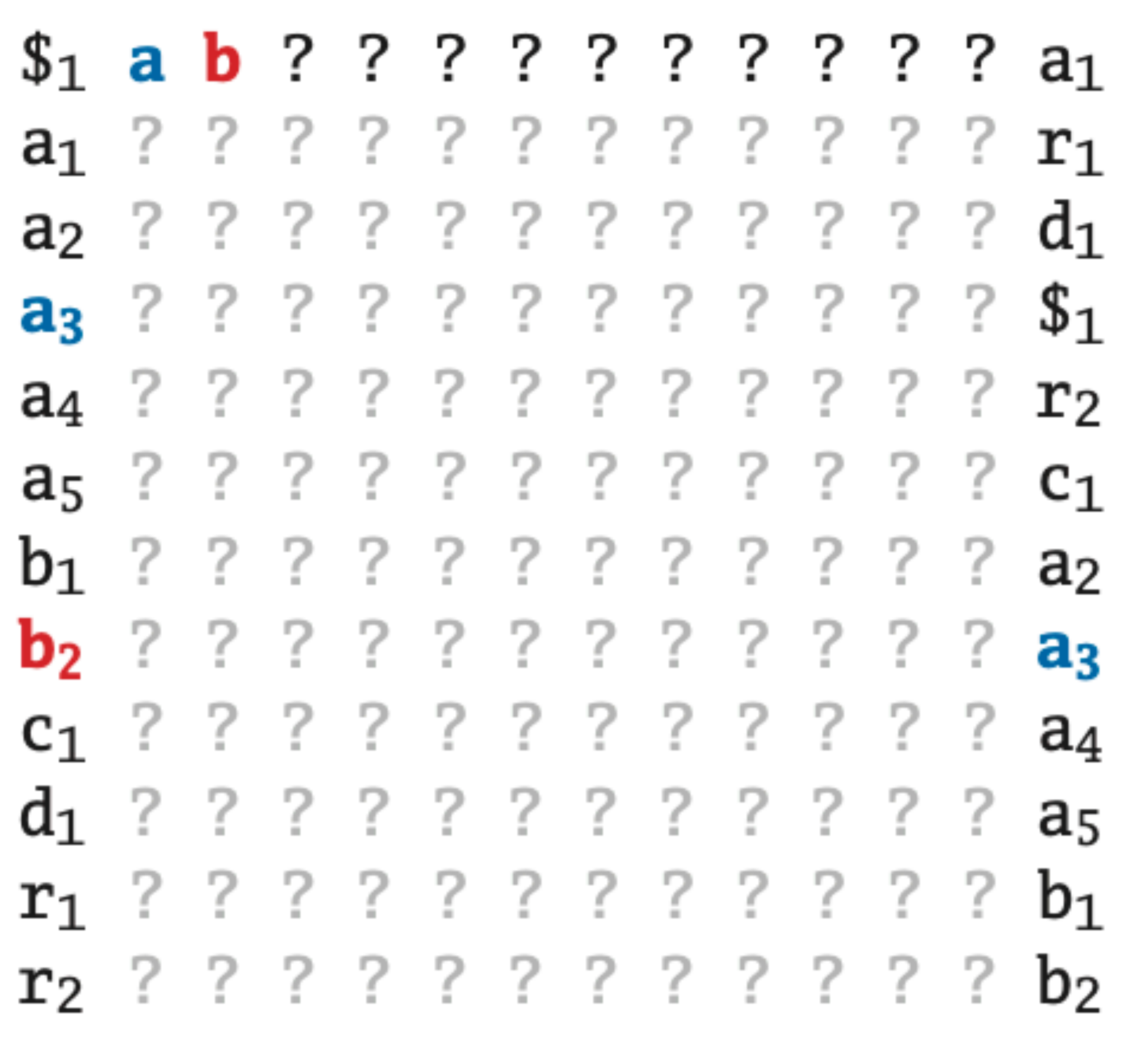


\$	p	a	n	a	m	a	b	a	n	a	n	a	s
a_1	b	a	n	a	n	a	s	\$	p	a	n	a	m
a_2	m	a	b	a	n	a	n	a	s	\$	p	a	n
a_3	n	a	m	a	b	a	n	a	n	a	s	\$	p
a_4	n	a	n	a	s	\$	p	a	n	a	m	a	b
a_5	n	a	s	\$	p	a	n	a	m	a	b	a	n
a_6	s	\$	p	a	n	a	m	a	b	a	n	a	n
b	a	n	a	n	a	s	\$	p	a	n	a	m	a
m	a	b	a	n	a	n	a	s	\$	p	a	n	a
n	a	m	a	b	a	n	a	n	a	s	\$	p	a
n	a	n	a	s	\$	p	a	n	a	m	a	b	a
n	a	s	\$	p	a	n	a	m	a	b	a	n	a
p	a	n	a	m	a	b	a	n	a	n	a	s	\$
s	\$	p	a	n	a	m	a	b	a	n	a	n	a

First-Last Property

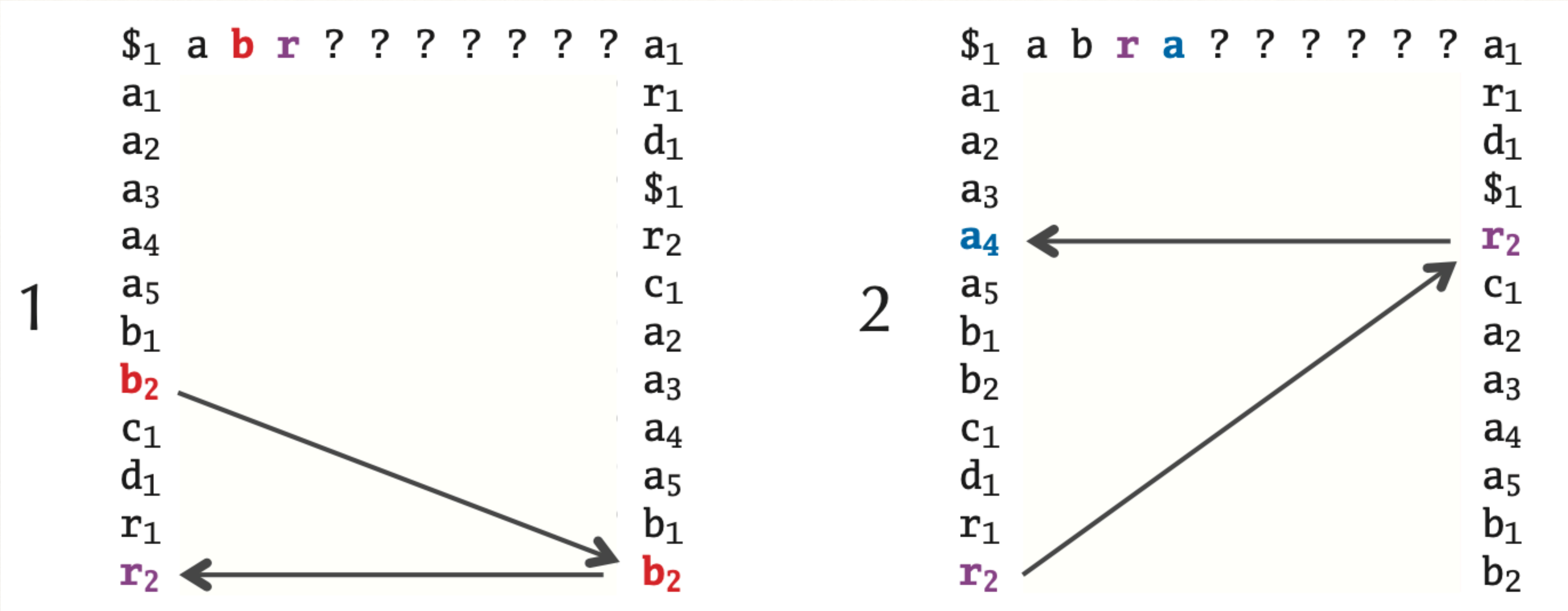


$\$_1$	a	?	?	?	?	?	?	?	?	?	?	?	a_1
a_1	?	?	?	?	?	?	?	?	?	?	?	?	r_1
a_2	?	?	?	?	?	?	?	?	?	?	?	?	d_1
a_3	?	?	?	?	?	?	?	?	?	?	?	?	$\$_1$
a_4	?	?	?	?	?	?	?	?	?	?	?	?	r_2
a_5	?	?	?	?	?	?	?	?	?	?	?	?	c_1
b_1	?	?	?	?	?	?	?	?	?	?	?	?	a_2
b_2	?	?	?	?	?	?	?	?	?	?	?	?	a_3
c_1	?	?	?	?	?	?	?	?	?	?	?	?	a_4
d_1	?	?	?	?	?	?	?	?	?	?	?	?	a_5
r_1	?	?	?	?	?	?	?	?	?	?	?	?	b_1
r_2	?	?	?	?	?	?	?	?	?	?	?	?	b_2



$\$_1$	a	b	?	?	?	?	?	?	?	?	?	?	a_1
a_1	?	?	?	?	?	?	?	?	?	?	?	?	r_1
a_2	?	?	?	?	?	?	?	?	?	?	?	?	d_1
a_3	?	?	?	?	?	?	?	?	?	?	?	?	$\$_1$
a_4	?	?	?	?	?	?	?	?	?	?	?	?	r_2
a_5	?	?	?	?	?	?	?	?	?	?	?	?	c_1
b_1	?	?	?	?	?	?	?	?	?	?	?	?	a_2
b_2	?	?	?	?	?	?	?	?	?	?	?	?	a_3
c_1	?	?	?	?	?	?	?	?	?	?	?	?	a_4
d_1	?	?	?	?	?	?	?	?	?	?	?	?	a_5
r_1	?	?	?	?	?	?	?	?	?	?	?	?	b_1
r_2	?	?	?	?	?	?	?	?	?	?	?	?	b_2

First-Last Property



1

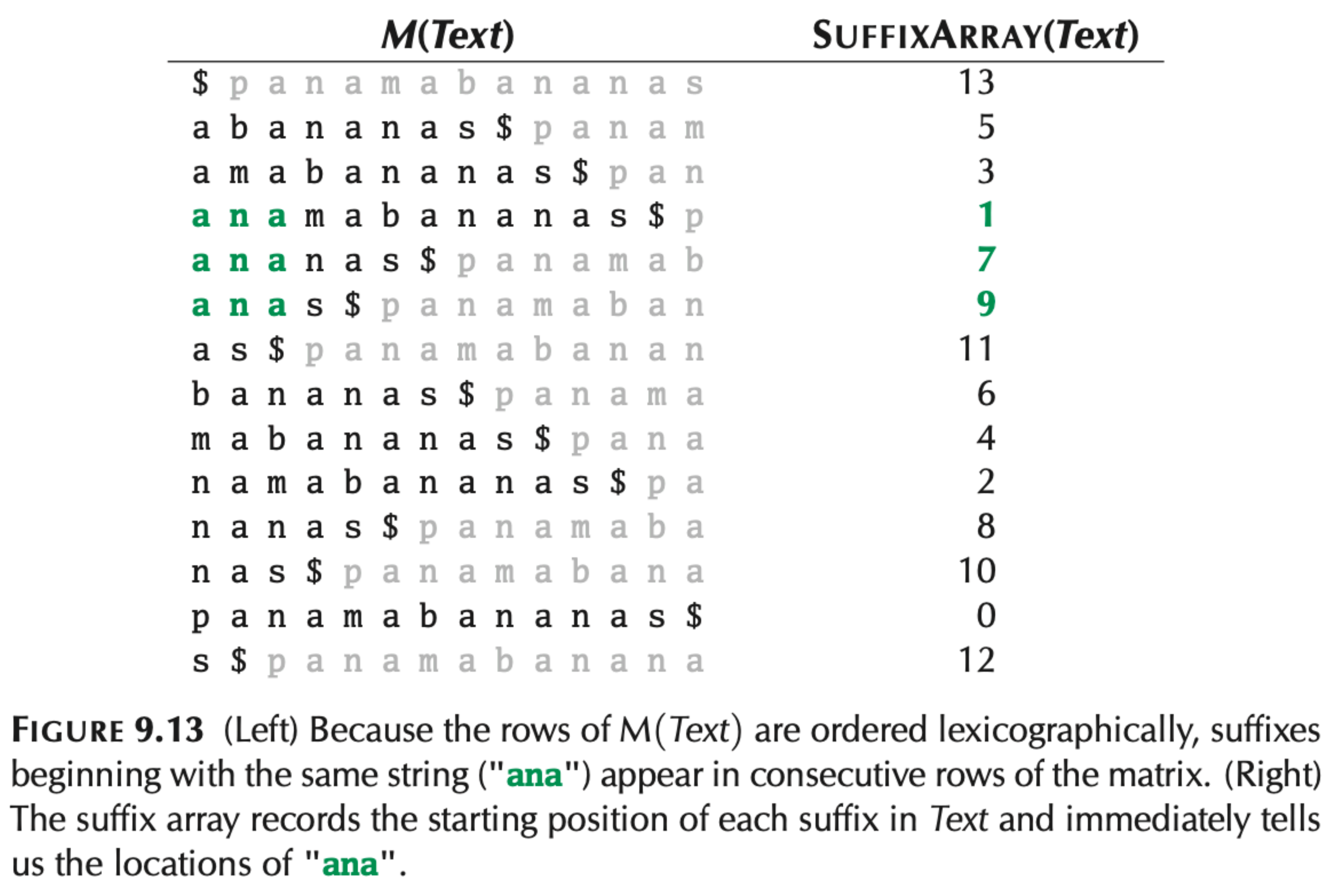
$\$_1$ a b r ? ? ? ? ? ? ? a_1
a_1 r_1
a_2 d_1
a_3 $\$_1$
a_4 r_2
a_5 c_1
b_1 a_2
b_2 a_3
c_1 a_4
d_1 a_5
r_1 b_1
r_2 b_2

2

$\$_1$ a b r a ? ? ? ? ? ? a_1
a_1 r_1
a_2 d_1
a_3 $\$_1$
a_4 r_2
a_5 c_1
b_1 a_2
b_2 a_3
c_1 a_4
d_1 a_5
r_1 b_1
r_2 b_2

Pattern Matching

- Each row of M(Text) begins with a different suffix of Text.



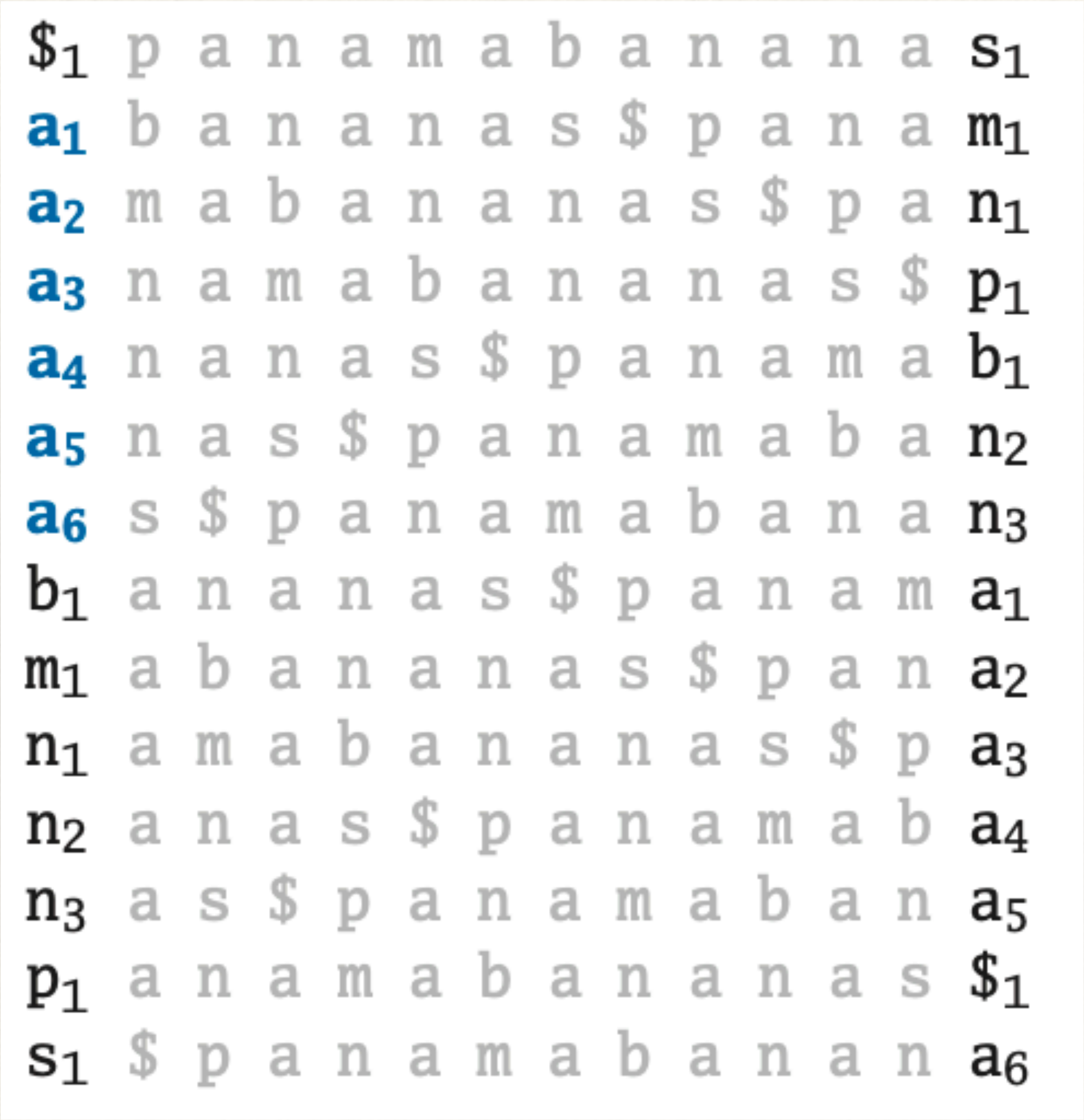
M(*Text*)	SUFFIXARRAY(*Text*)
$panamabananas	13
abananas$panam	5
amabananas$pan	3
anamabananas$p	**1**
ananas$panamab	**7**
anas$panamaban	**9**
as$panamabanan	11
bananas$panama	6
mabananas$pana	4
namabananas$pa	2
nanas$panamaba	8
nas$panamabana	10
panamabananas$	0
s$panamabanana	12

FIGURE 9.13 (Left) Because the rows of M(*Text*) are ordered lexicographically, suffixes beginning with the same string ("**ana**") appear in consecutive rows of the matrix. (Right) The suffix array records the starting position of each suffix in *Text* and immediately tells us the locations of "**ana**".

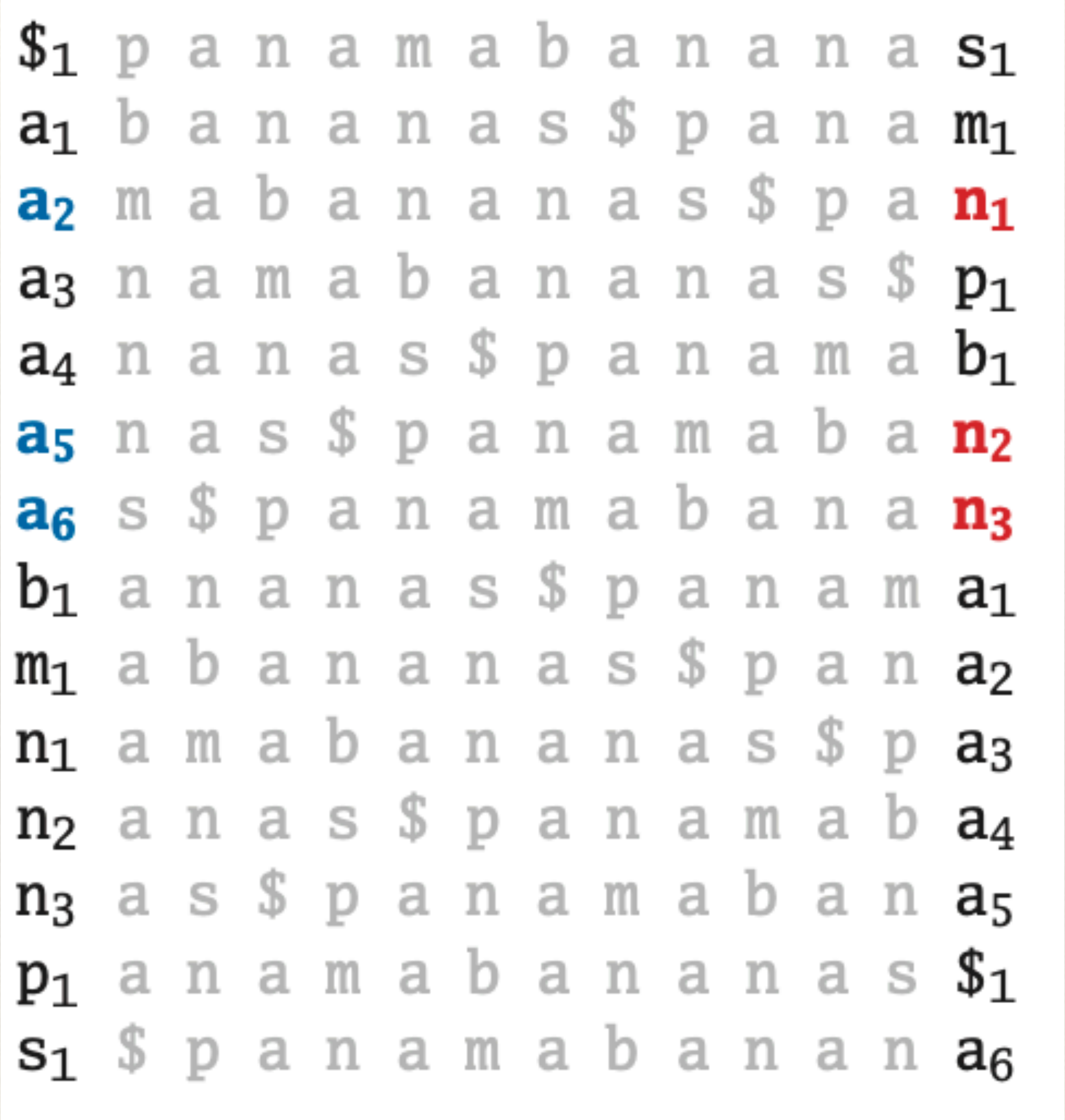
Problems?

- We cannot afford storing the entire matrix M(Text), which has |Text|2 entries.
- Let's forbid ourselves from accessing any information in M(Text) other than FirstColumn and LastColumn.
- Using these two columns, we will try to match Pattern to Text by moving backward through Pattern.

Pattern Matching

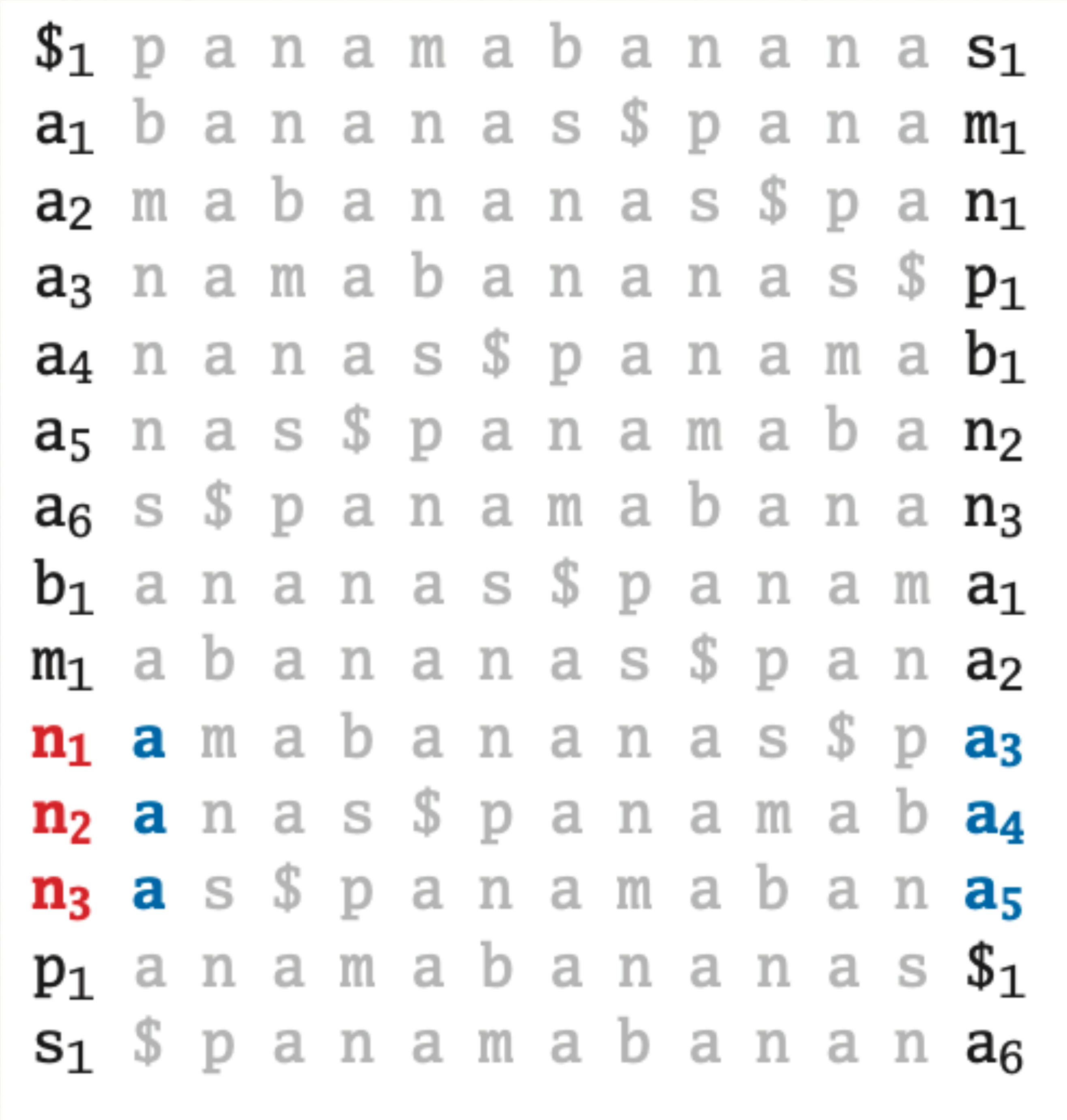


$\$_1$	p	a	n	a	m	a	b	a	n	a	n	a	s_1
a_1	b	a	n	a	n	a	s	$\$$	p	a	n	a	m_1
a_2	m	a	b	a	n	a	n	a	s	$\$$	p	a	n_1
a_3	n	a	m	a	b	a	n	a	n	a	s	$\$$	p_1
a_4	n	a	n	a	s	$\$$	p	a	n	a	m	a	b_1
a_5	n	a	s	$\$$	p	a	n	a	m	a	b	a	n_2
a_6	s	$\$$	p	a	n	a	m	a	b	a	n	a	n_3
b_1	a	n	a	n	a	s	$\$$	p	a	n	a	m	a_1
m_1	a	b	a	n	a	n	a	s	$\$$	p	a	n	a_2
n_1	a	m	a	b	a	n	a	n	a	s	$\$$	p	a_3
n_2	a	n	a	s	$\$$	p	a	n	a	m	a	b	a_4
n_3	a	s	$\$$	p	a	n	a	m	a	b	a	n	a_5
p_1	a	n	a	m	a	b	a	n	a	n	a	s	$\$_1$
s_1	$\$$	p	a	n	a	m	a	b	a	n	a	n	a_6

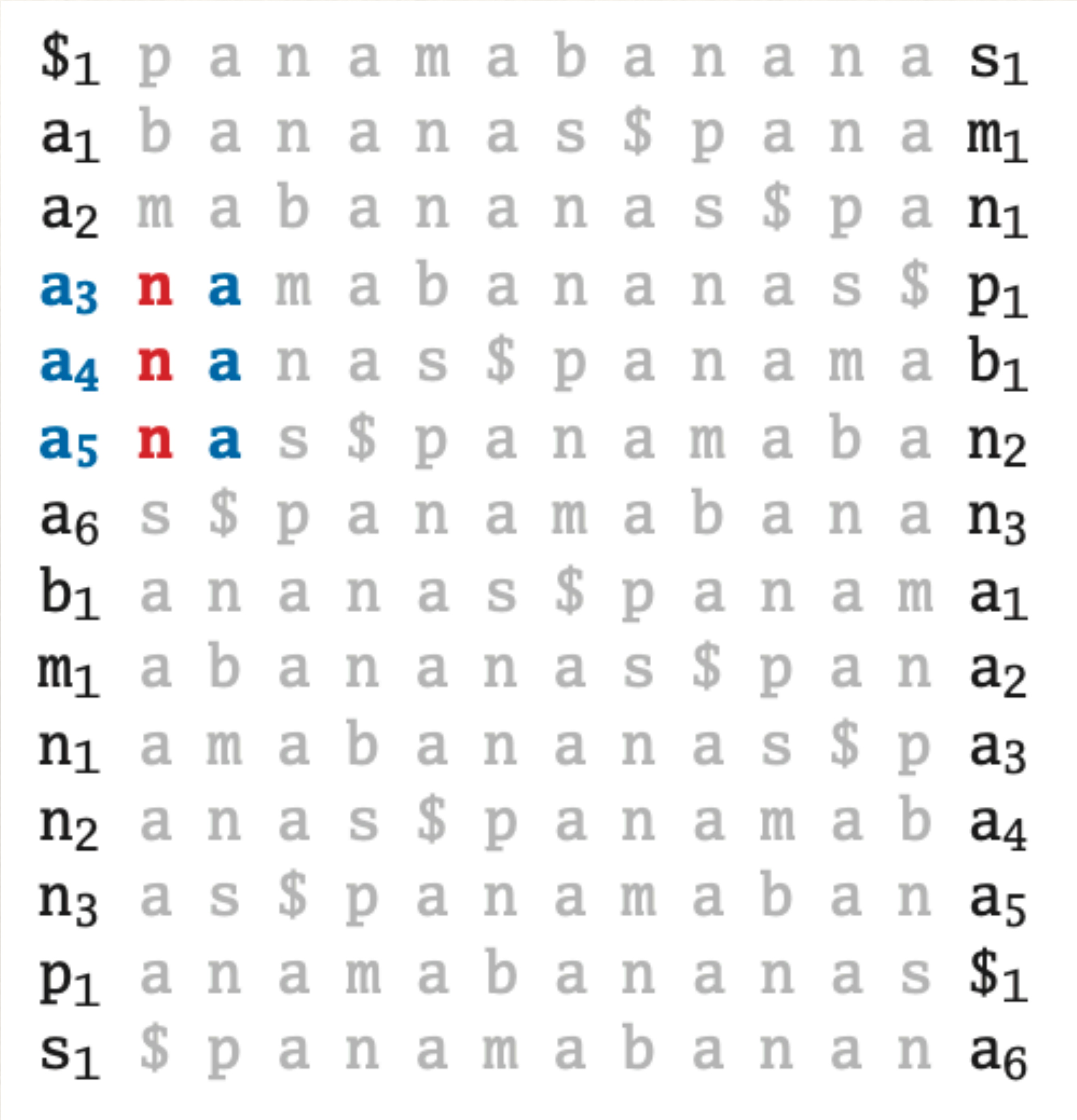


$\$_1$	p	a	n	a	m	a	b	a	n	a	n	a	s_1
a_1	b	a	n	a	n	a	s	$\$$	p	a	n	a	m_1
a_2	m	a	b	a	n	a	n	a	s	$\$$	p	a	**n_1**
a_3	n	a	m	a	b	a	n	a	n	a	s	$\$$	p_1
a_4	n	a	n	a	s	$\$$	p	a	n	a	m	a	b_1
a_5	n	a	s	$\$$	p	a	n	a	m	a	b	a	**n_2**
a_6	s	$\$$	p	a	n	a	m	a	b	a	n	a	**n_3**
b_1	a	n	a	n	a	s	$\$$	p	a	n	a	m	a_1
m_1	a	b	a	n	a	n	a	s	$\$$	p	a	n	a_2
n_1	a	m	a	b	a	n	a	n	a	s	$\$$	p	a_3
n_2	a	n	a	s	$\$$	p	a	n	a	m	a	b	a_4
n_3	a	s	$\$$	p	a	n	a	m	a	b	a	n	a_5
p_1	a	n	a	m	a	b	a	n	a	n	a	s	$\$_1$
s_1	$\$$	p	a	n	a	m	a	b	a	n	a	n	a_6

Pattern Matching



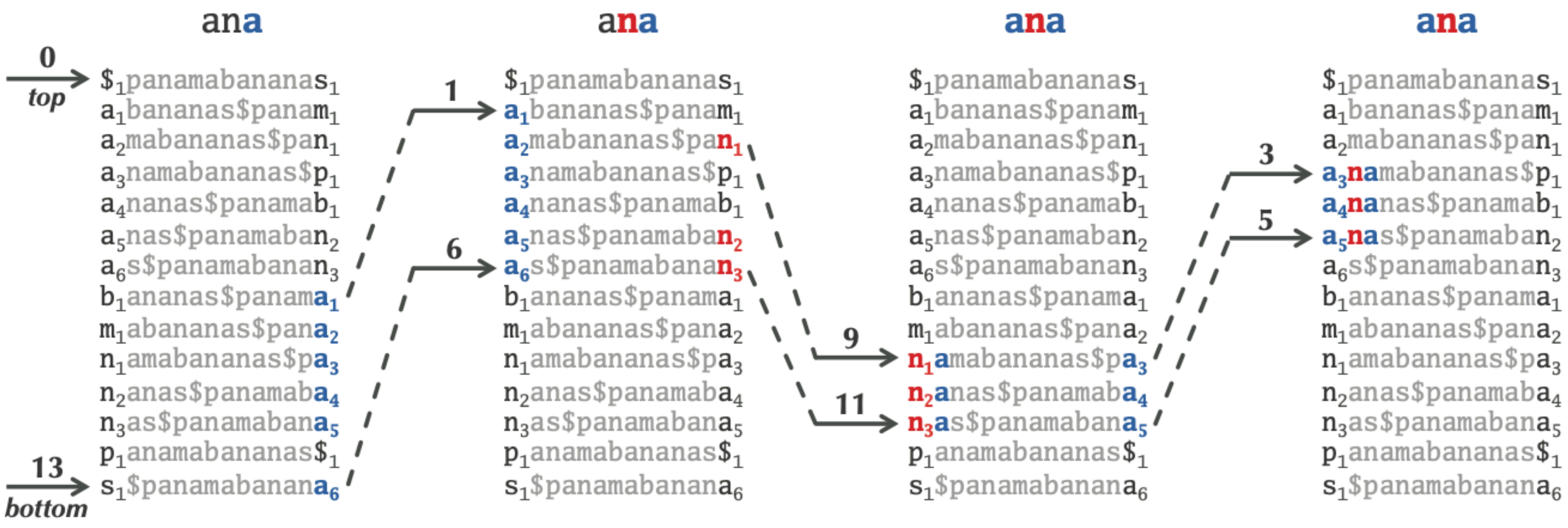
$\$_1$	p	a	n	a	m	a	b	a	n	a	n	a	s_1
a_1	b	a	n	a	n	a	s	$	p	a	n	a	m_1
a_2	m	a	b	a	n	a	n	a	s	$	p	a	n_1
a_3	n	a	m	a	b	a	n	a	n	a	s	$	p_1
a_4	n	a	n	a	s	$	p	a	n	a	m	a	b_1
a_5	n	a	s	$	p	a	n	a	m	a	b	a	n_2
a_6	s	$	p	a	n	a	m	a	b	a	n	a	n_3
b_1	a	n	a	n	a	s	$	p	a	n	a	m	a_1
m_1	a	b	a	n	a	n	a	s	$	p	a	n	a_2
n_1	**a**	m	a	b	a	n	a	n	a	s	$	p	**a_3**
n_2	**a**	n	a	s	$	p	a	n	a	m	a	b	**a_4**
n_3	**a**	s	$	p	a	n	a	m	a	b	a	n	**a_5**
p_1	a	n	a	m	a	b	a	n	a	n	a	s	$\$_1$
s_1	$	p	a	n	a	m	a	b	a	n	a	n	a_6



$\$_1$	p	a	n	a	m	a	b	a	n	a	n	a	s_1
a_1	b	a	n	a	n	a	s	$	p	a	n	a	m_1
a_2	m	a	b	a	n	a	n	a	s	$	p	a	n_1
a_3	**n**	**a**	m	a	b	a	n	a	n	a	s	$	p_1
a_4	**n**	**a**	n	a	s	$	p	a	n	a	m	a	b_1
a_5	**n**	**a**	s	$	p	a	n	a	m	a	b	a	n_2
a_6	s	$	p	a	n	a	m	a	b	a	n	a	n_3
b_1	a	n	a	n	a	s	$	p	a	n	a	m	a_1
m_1	a	b	a	n	a	n	a	s	$	p	a	n	a_2
n_1	a	m	a	b	a	n	a	n	a	s	$	p	a_3
n_2	a	n	a	s	$	p	a	n	a	m	a	b	a_4
n_3	a	s	$	p	a	n	a	m	a	b	a	n	a_5
p_1	a	n	a	m	a	b	a	n	a	n	a	s	$\$_1$
s_1	$	p	a	n	a	m	a	b	a	n	a	n	a_6

Pattern Matching

- We know that at each step, the rows of M(Text) that match a suffix of Pattern clump together in consecutive rows of M(Text).



ana
ana
ana
ana
0
top
13
bottom
1
6
9
11
3
5
$\$_1\mathtt{panamabanana}s_1$
$a_1\mathtt{bananas\$pana}m_1$
$a_2\mathtt{mabananas\$pa}n_1$
$a_3\mathtt{namabananas\$}p_1$
$a_4\mathtt{nanas\$panama}b_1$
$a_5\mathtt{nas\$panamaba}n_2$
$a_6\mathtt{s\$panamabana}n_3$
$b_1\mathtt{ananas\$panam}a_1$
$m_1\mathtt{abananas\$pan}a_2$
$n_1\mathtt{amabananas\$p}a_3$
$n_2\mathtt{anas\$panamab}a_4$
$n_3\mathtt{as\$panamaban}a_5$
$p_1\mathtt{anamabananas}\$_1$
$s_1\mathtt{\$panamabanan}a_6$

FIGURE 9.14 The pointers *top* and *bottom* hold the indices of the first and last rows of M(*Text*) matching the current suffix of *Pattern* = "ana". The above diagram shows how these pointers are updated when walking backwards through "ana" and looking for substring matches in "panamabananas$".