Burrows-Wheeler Transform and Suffix Arrays

Pavel Pevzner

Department of Computer Science and Engineering
University of California at San Diego

Algorithms on Strings
Data Structures and Algorithms

This slide desk is incomplete.
For the complete set of frames,
please see our videos in the
Algorithms on Strings course on Coursera
(Algorithms and Data Structures Specialization)

Outline

- **Burrows-Wheeler Transform**
- Inverting Burrows-Wheeler Transform
- Using BWT for Pattern Matching
- Suffix Arrays
- Approximate Pattern Matching

Text Compression by Run-Length Encoding

- **Run-length encoding** compresses a run of *n* identical symbols:

Text

GGGGGGGGGGCCCCCCCCCCCAAAAAAATTTTTTTTTTTTTTTCCCCCG

↓

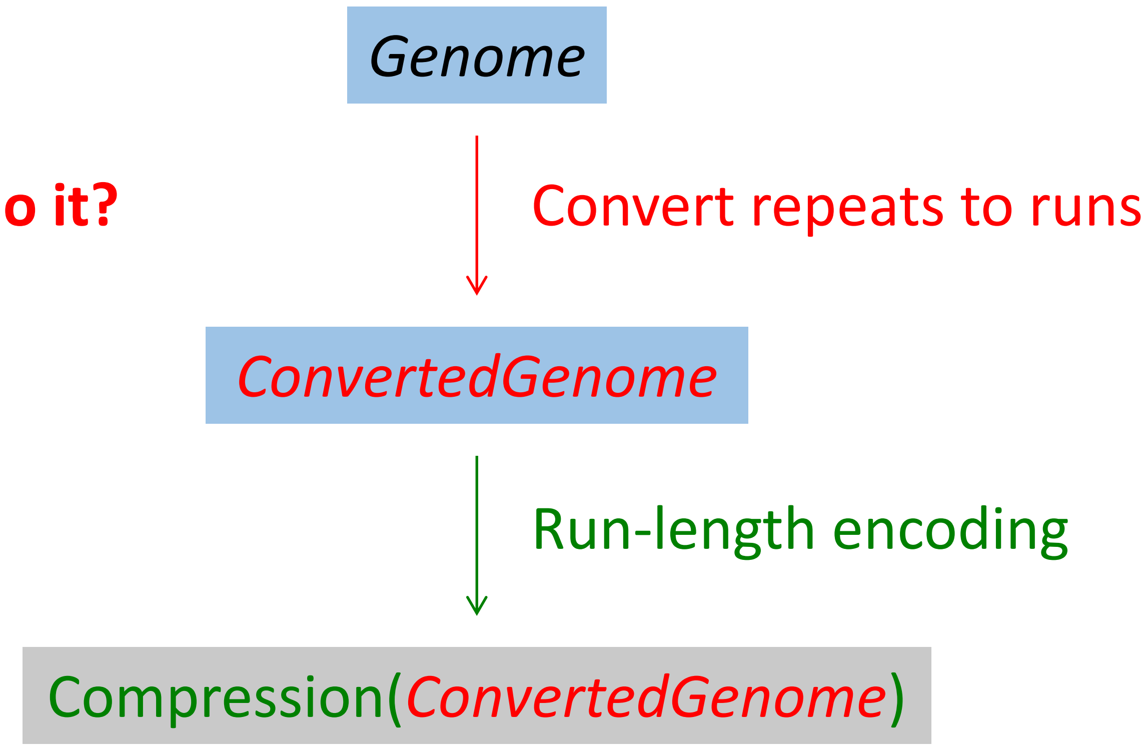
10G11C7A15T5C1G

- genomes don’t have lots of runs… but they do have lots of repeats:

ACTGACCGAAACTGAGTATCCGACTGAAACTGATCAGTACTGACATTGC

Idea: Converting Repeats to Runs

How do we do it?



Genome

Convert repeats to runs

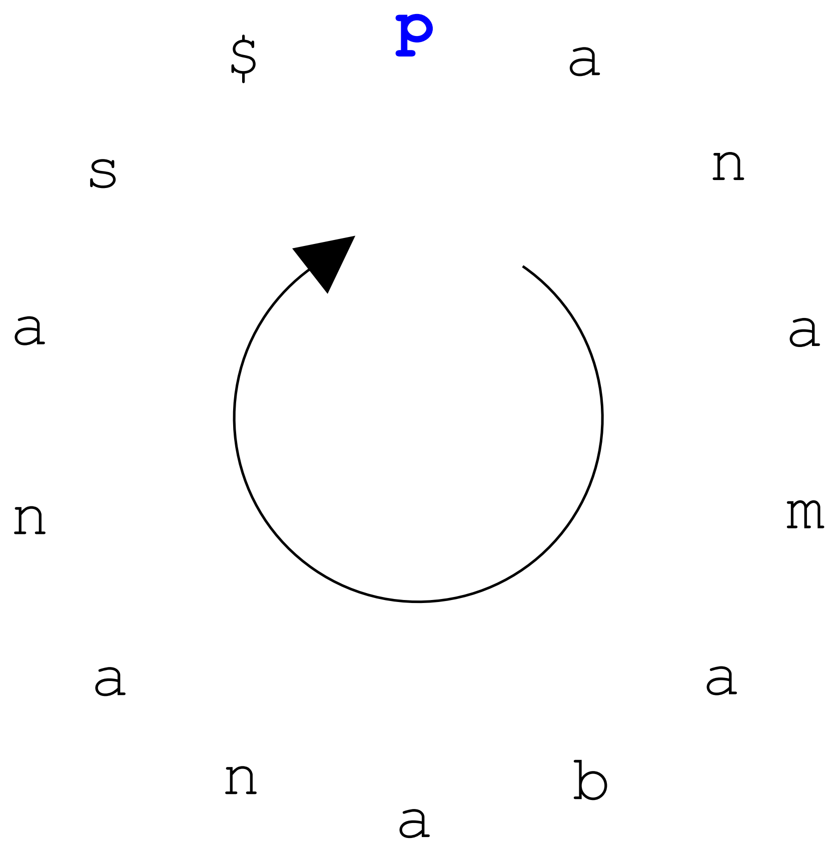
ConvertedGenome

Run-length encoding

Compression(*ConvertedGenome*)

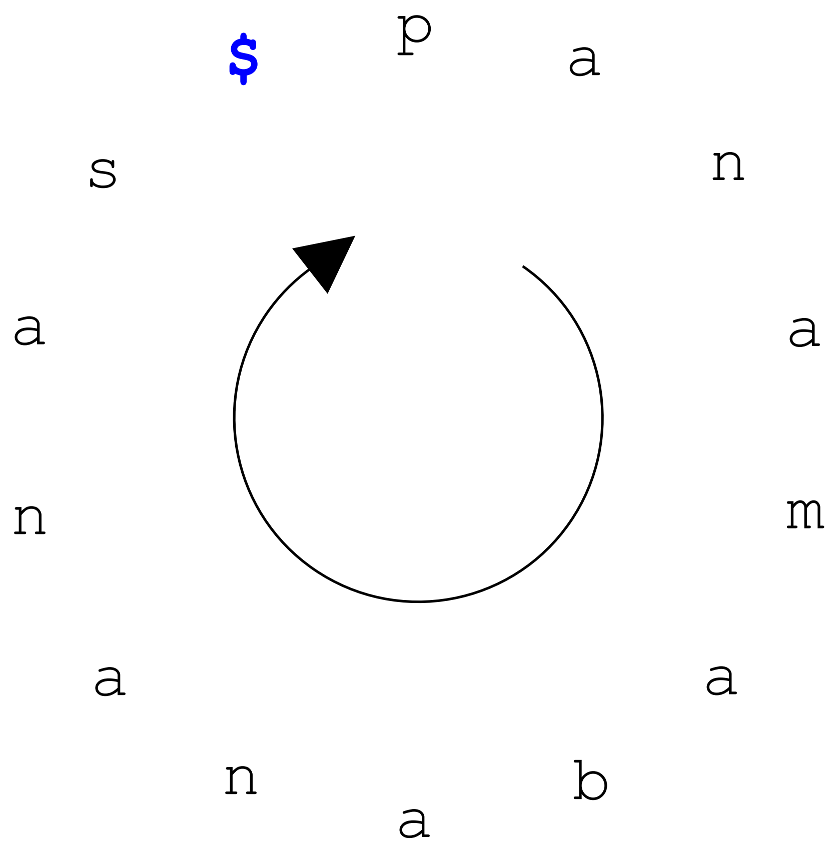
Forming All Cyclic Rotations of *Text*

panamabananas$



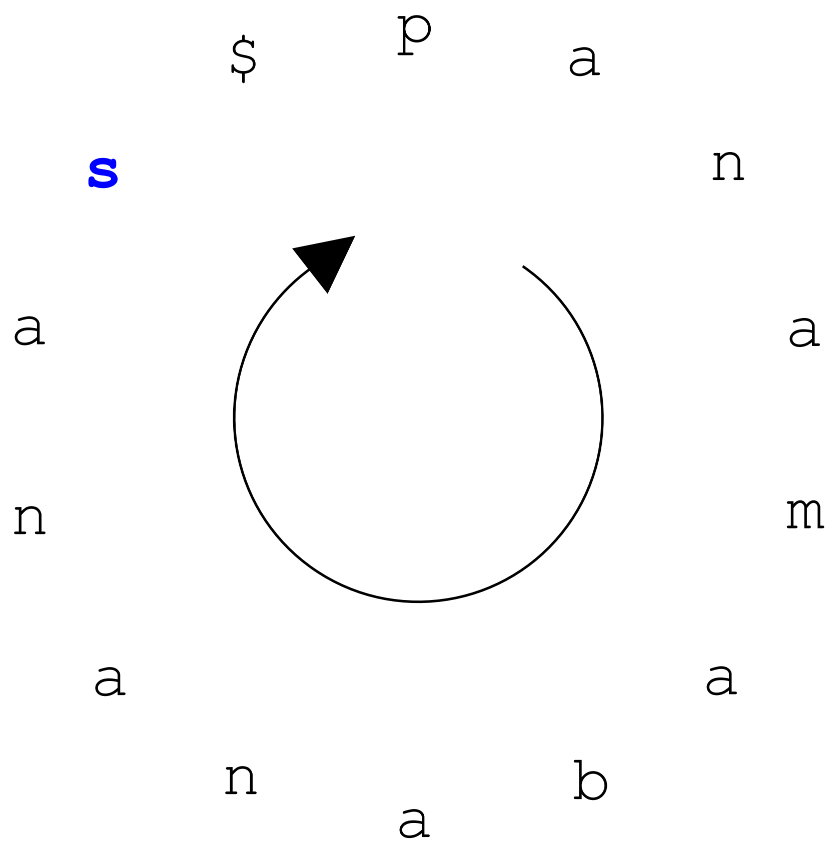
Cyclic Rotations

panamabananas$
$panamabananas



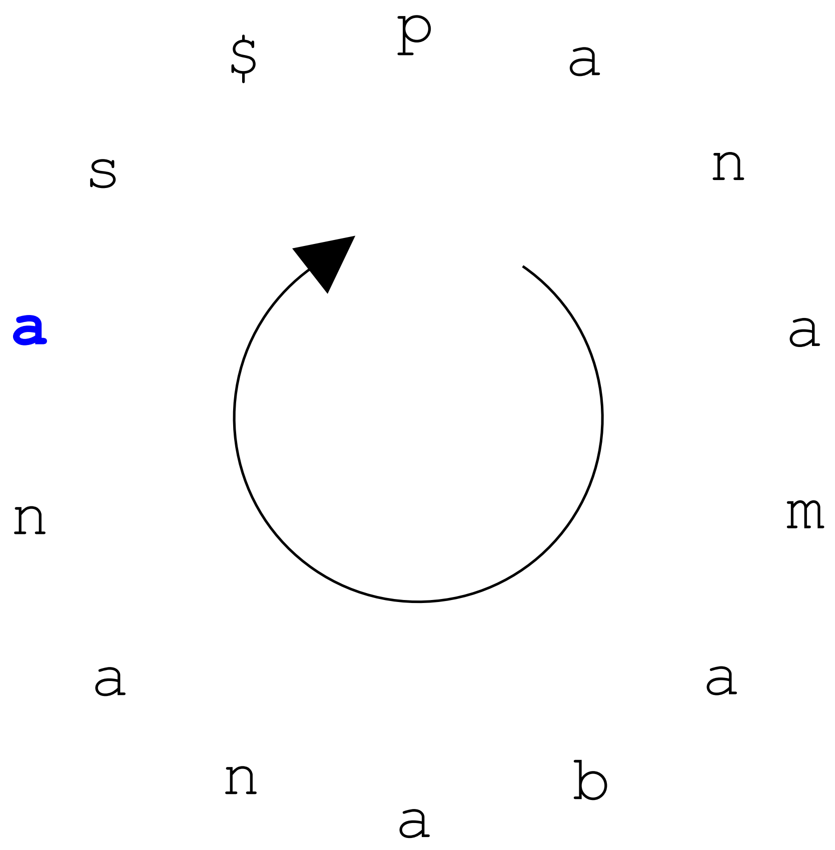
Cyclic Rotations

```
panamabananas$
$panamabananas
s$panamabanana
```



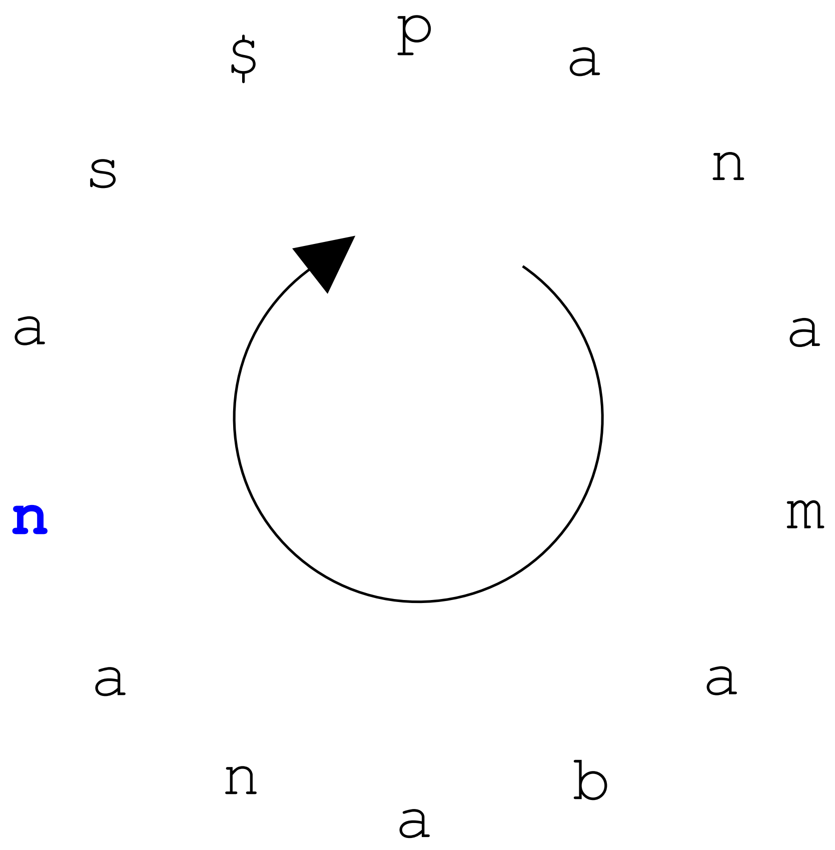
Cyclic Rotations

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
```



Cyclic Rotations

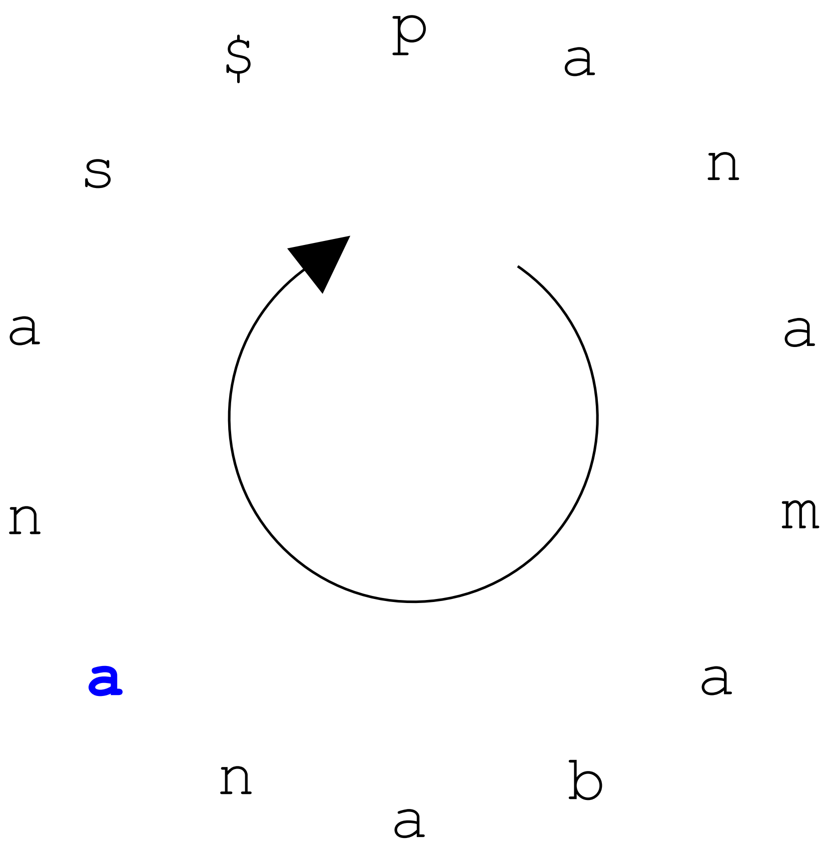
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana



Cyclic Rotations

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
```

anas$panamaban



Cyclic Rotations

panamabananas$

$panamabananas

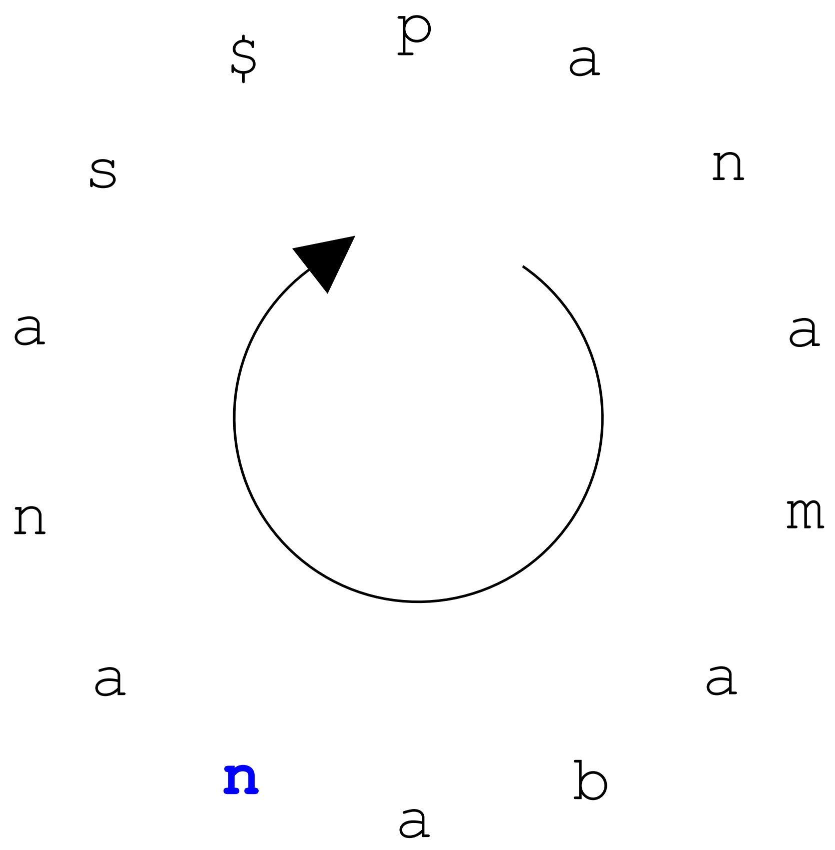
s$panamabanana

as$panamabanan

nas$panamabana

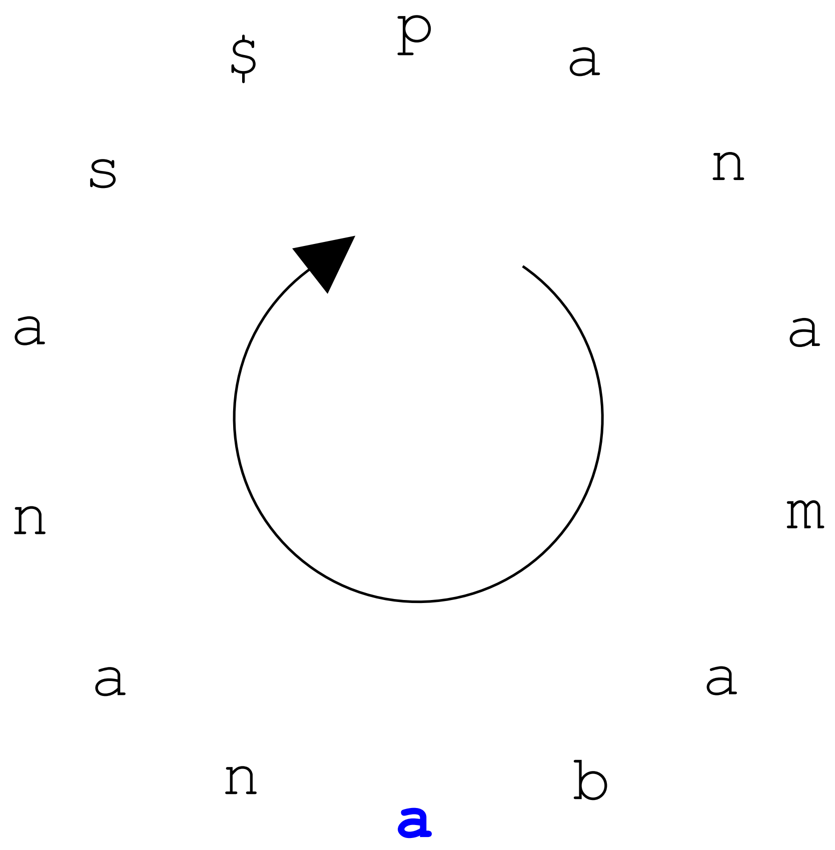
anas$panamaban

nanas$panamaba



Cyclic Rotations

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
```



Cyclic Rotations

panamabananas$

$panamabananas

s$panamabanana

as$panamabanan

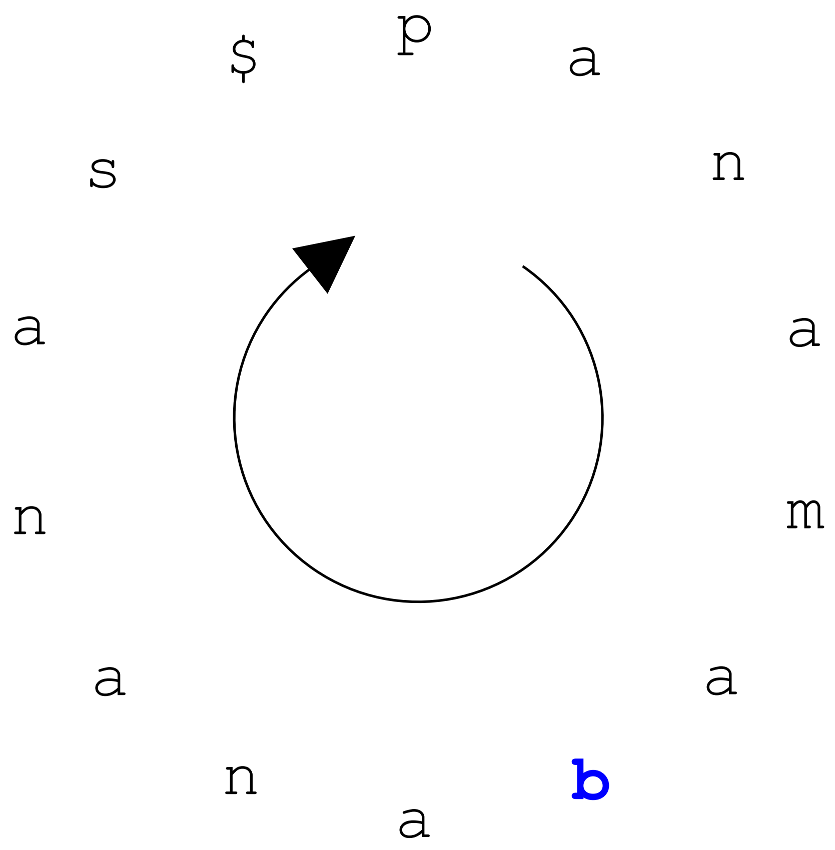
nas$panamabana

anas$panamaban

nanas$panamaba

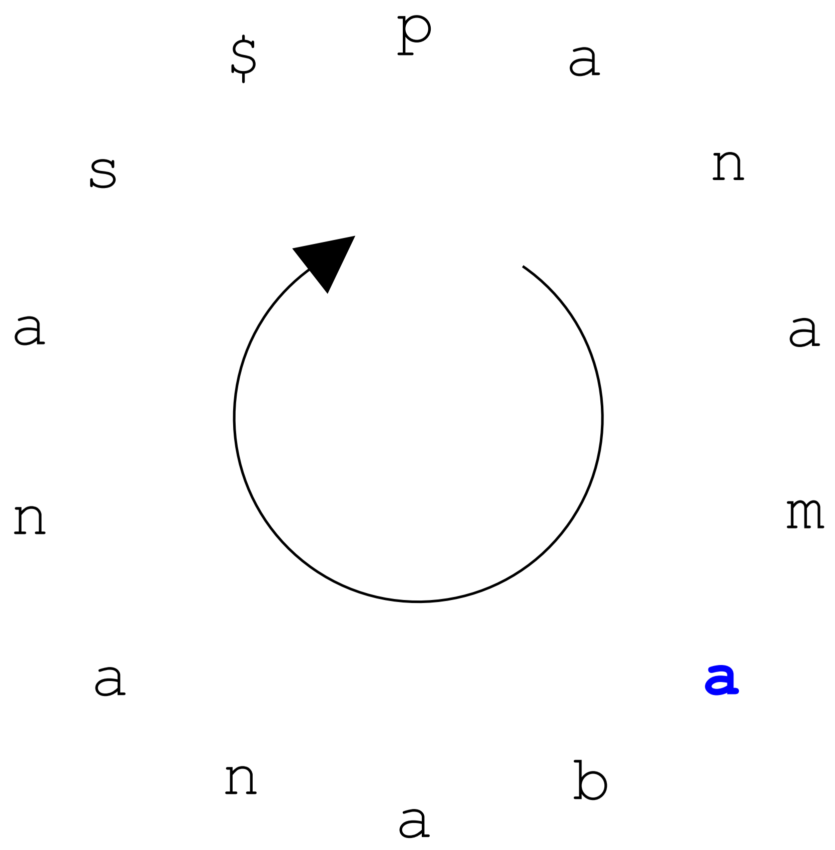
ananas$panamab

bananas$panama



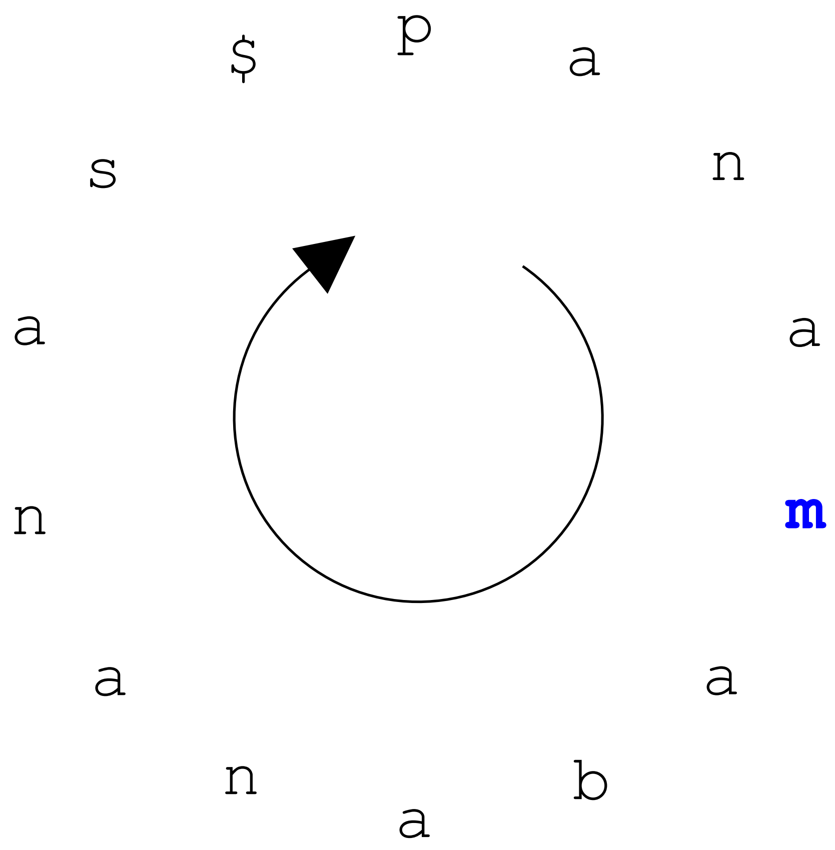
Cyclic Rotations

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
```



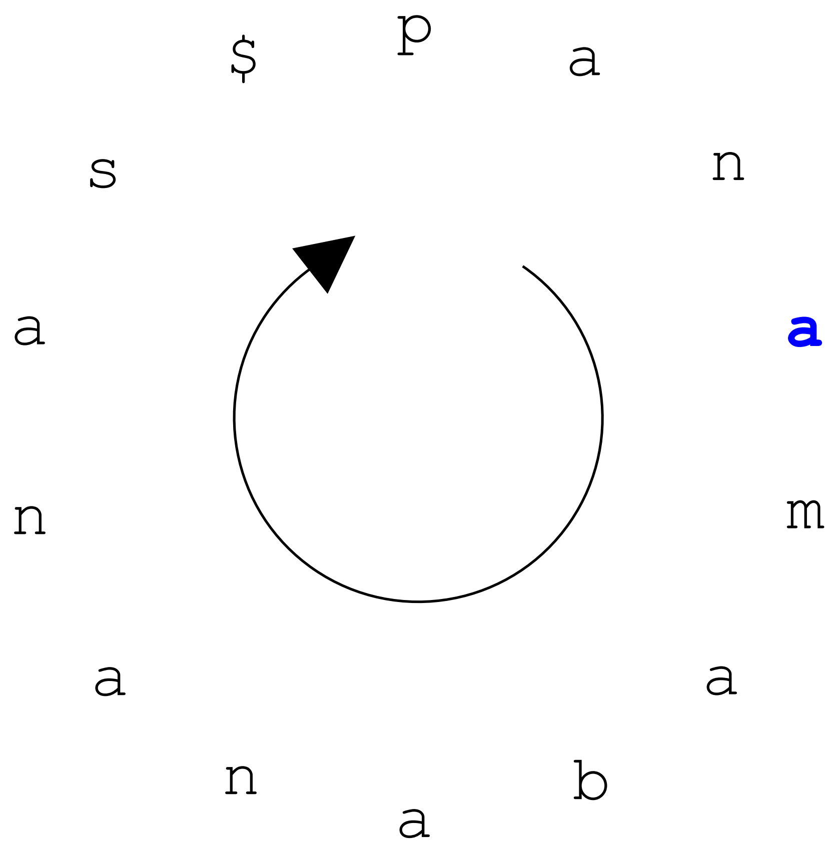
Cyclic Rotations

panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana



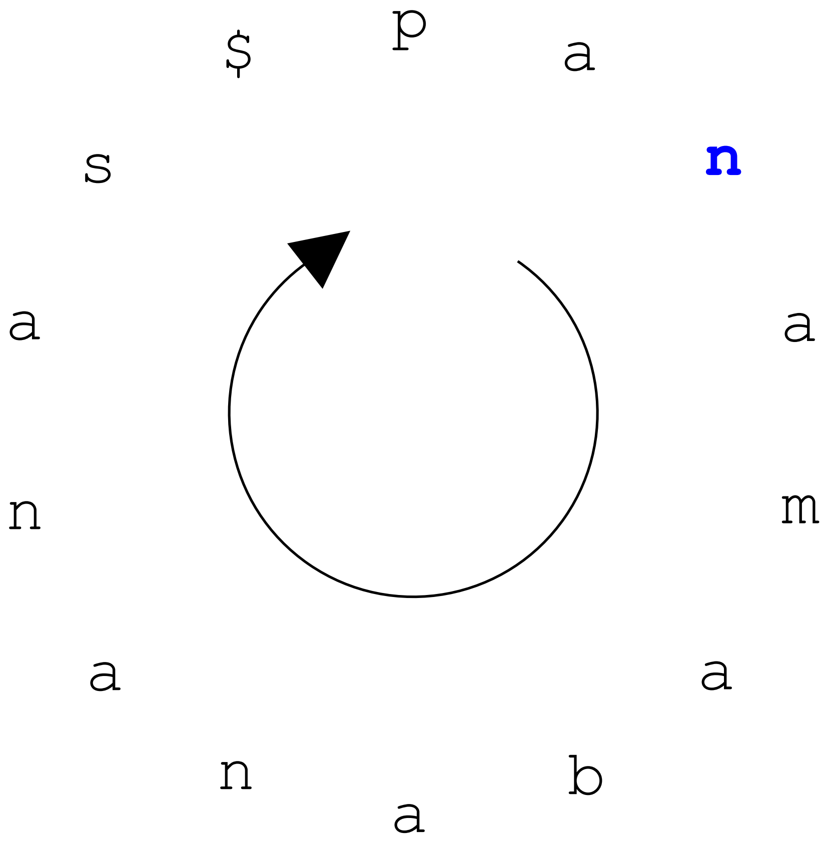
Cyclic Rotations

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana
amabananas$pan
```



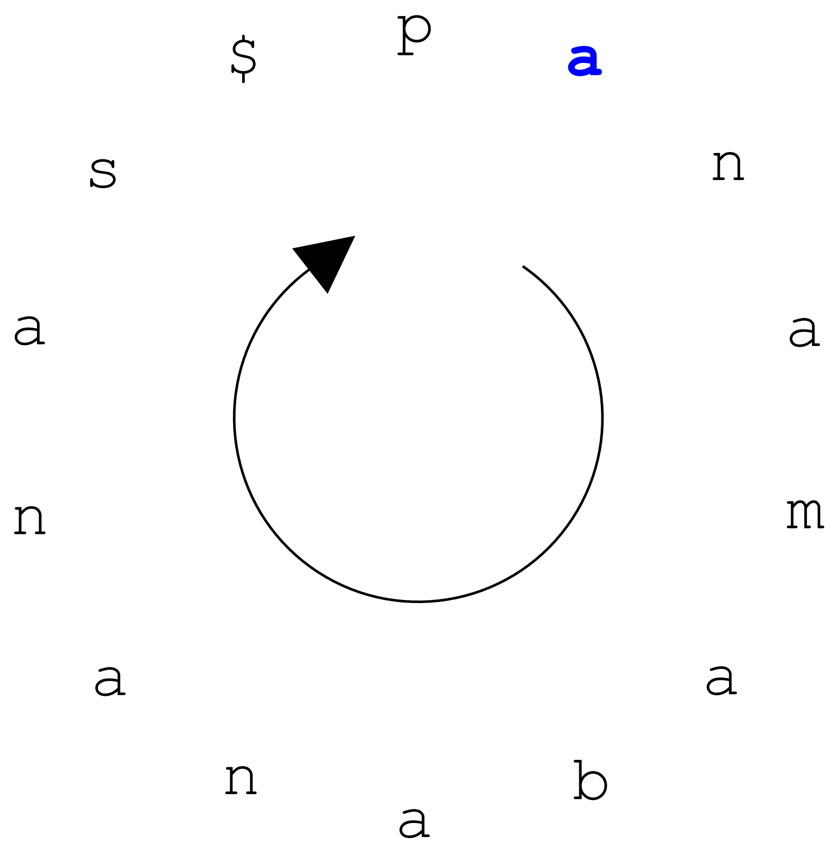
Cyclic Rotations

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana
amabananas$pan
namabananas$pa
```



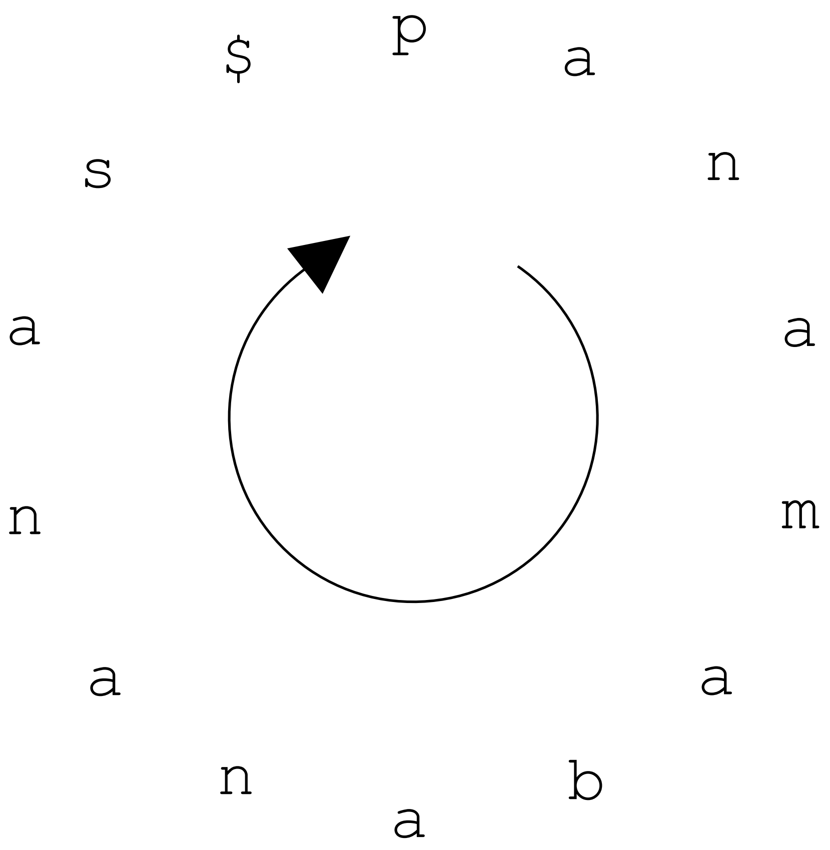
Cyclic Rotations

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana
amabananas$pan
namabananas$pa
anamabananas$p
```



Cyclic Rotations

panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana
amabananas$pan
namabananas$pa
anamabananas$p



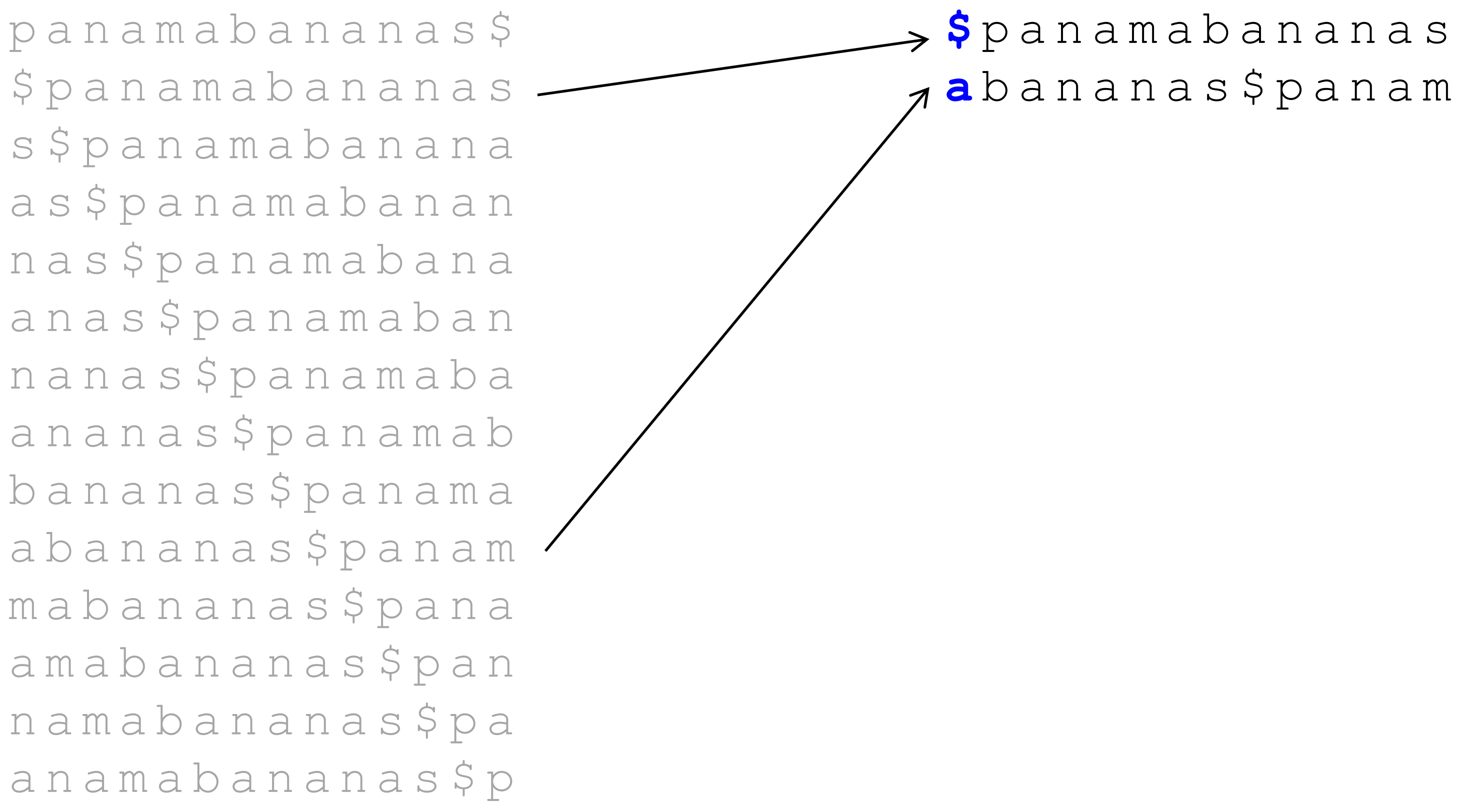
Sorting Cyclic Rotations

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana
amabananas$pan
namabananas$pa
anamabananas$p
```

```
$panamabananas
```

Sort the strings lexicographically ($ comes first)

Sorting Cyclic Rotations

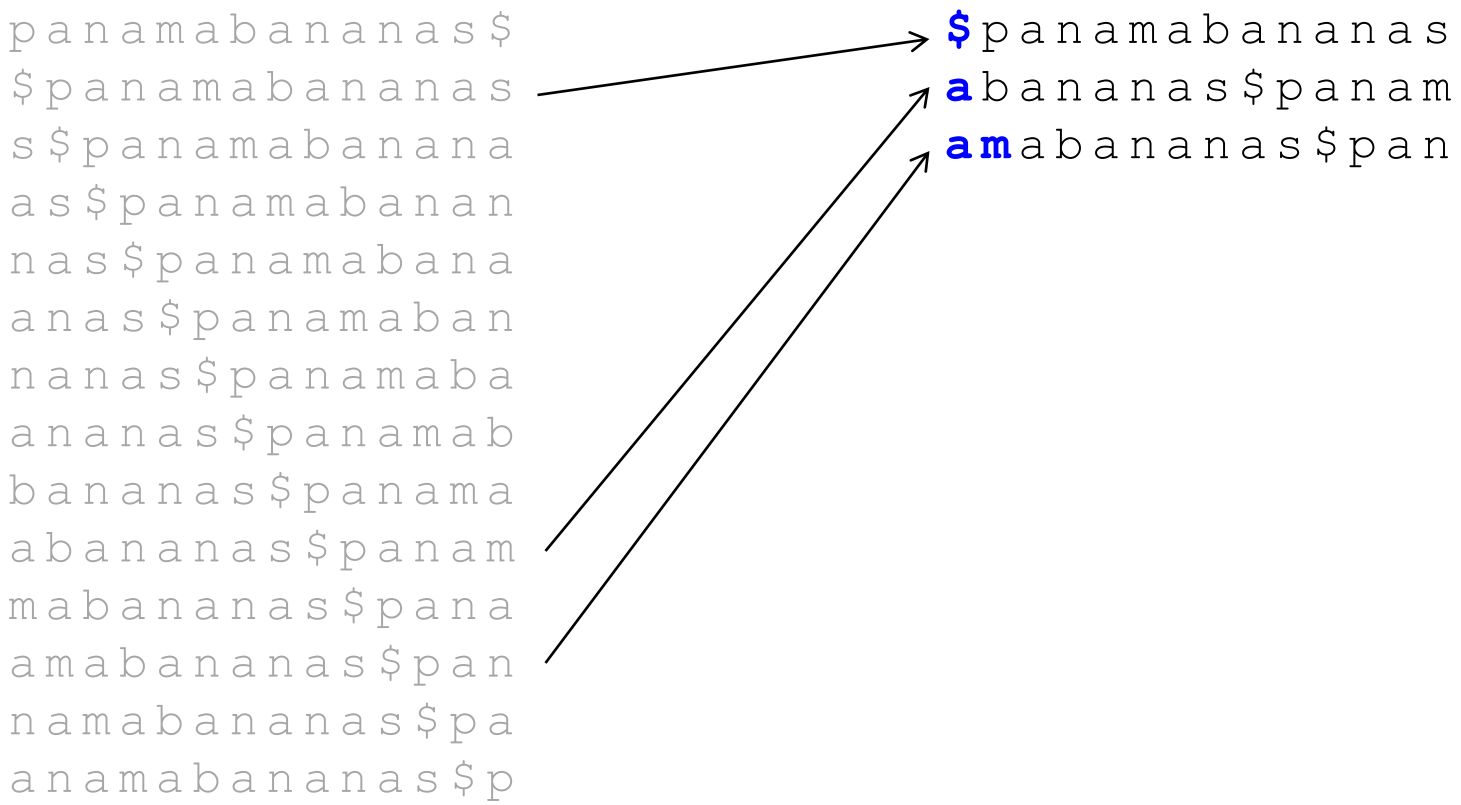


```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana
amabananas$pan
namabananas$pa
anamabananas$p
```

```
$panamabananas
abananas$panam
```

Sort the strings
lexicographically
($ comes first)

Sorting Cyclic Rotations

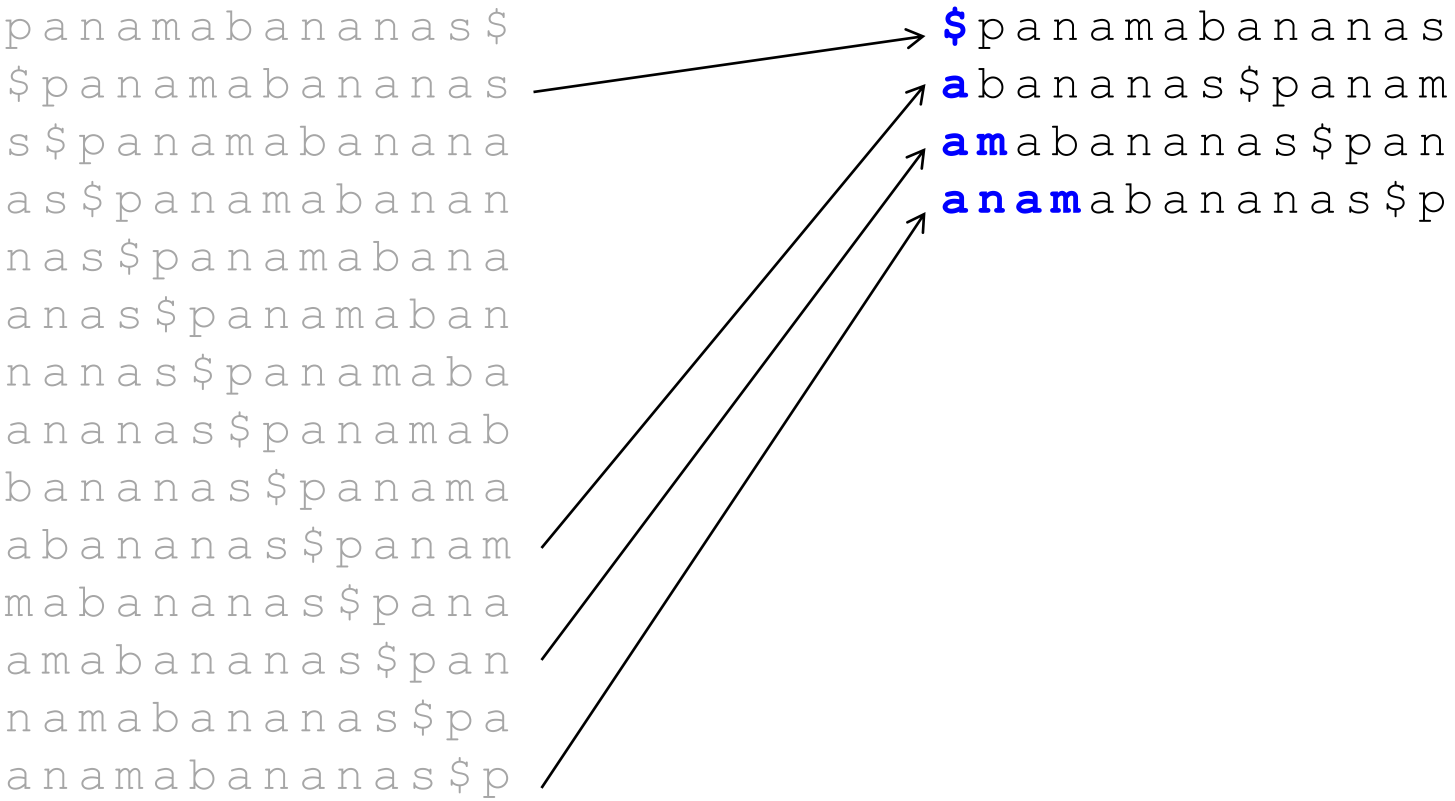


```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana
amabananas$pan
namabananas$pa
anamabananas$p
```

```
$panamabananas
abananas$panam
amabananas$pan
```

Sort the strings lexicographically ($ comes first)

Sorting Cyclic Rotations

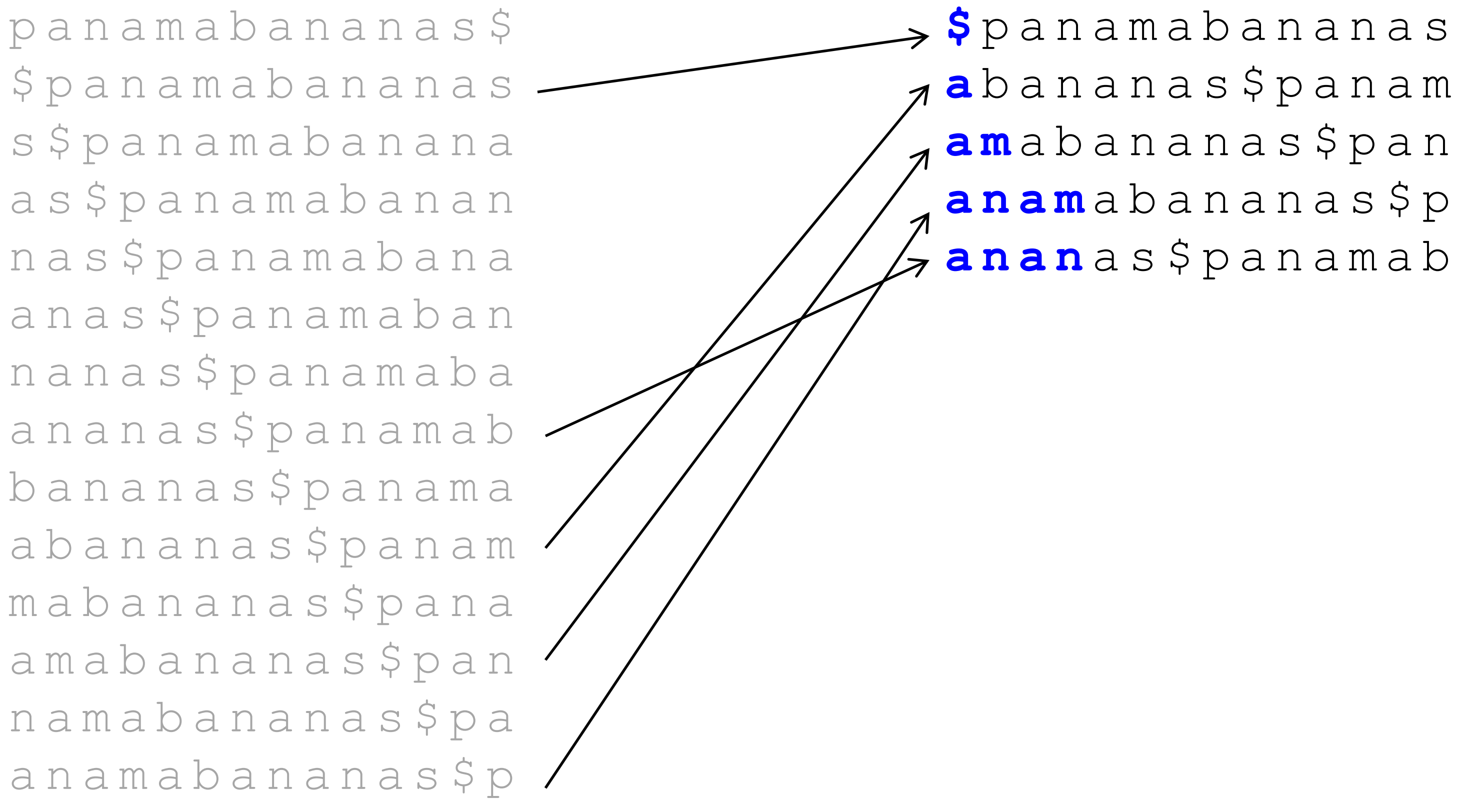


panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana
amabananas$pan
namabananas$pa
anamabananas$p

$panamabananas
abananas$panam
amabananas$pan
anamabananas$p

Sort the strings
lexicographically
($ comes first)

Sorting Cyclic Rotations



panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana
amabananas$pan
namabananas$pa
anamabananas$p

$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab

Sort the strings
lexicographically
($ comes first)

Sorting Cyclic Rotations

panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana
amabananas$pan
namabananas$pa
anamabananas$p

→

$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana

Sort the strings
lexicographically
($ comes first)

BWT(panamabananas$)=smnpbnnaaaaa$a

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**

All cyclic rotations of
“panamabananas$”

Burrows-Wheeler Transform (BWT):
Last column = **smnpbnnaaaaa$a**

BWT(panamabananas$)=smnpbnnaaaaa$a

```
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana
amabananas$pan
namabananas$pa
anamabananas$p
```

→

```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```

All cyclic rotations of
“panamabananas$”

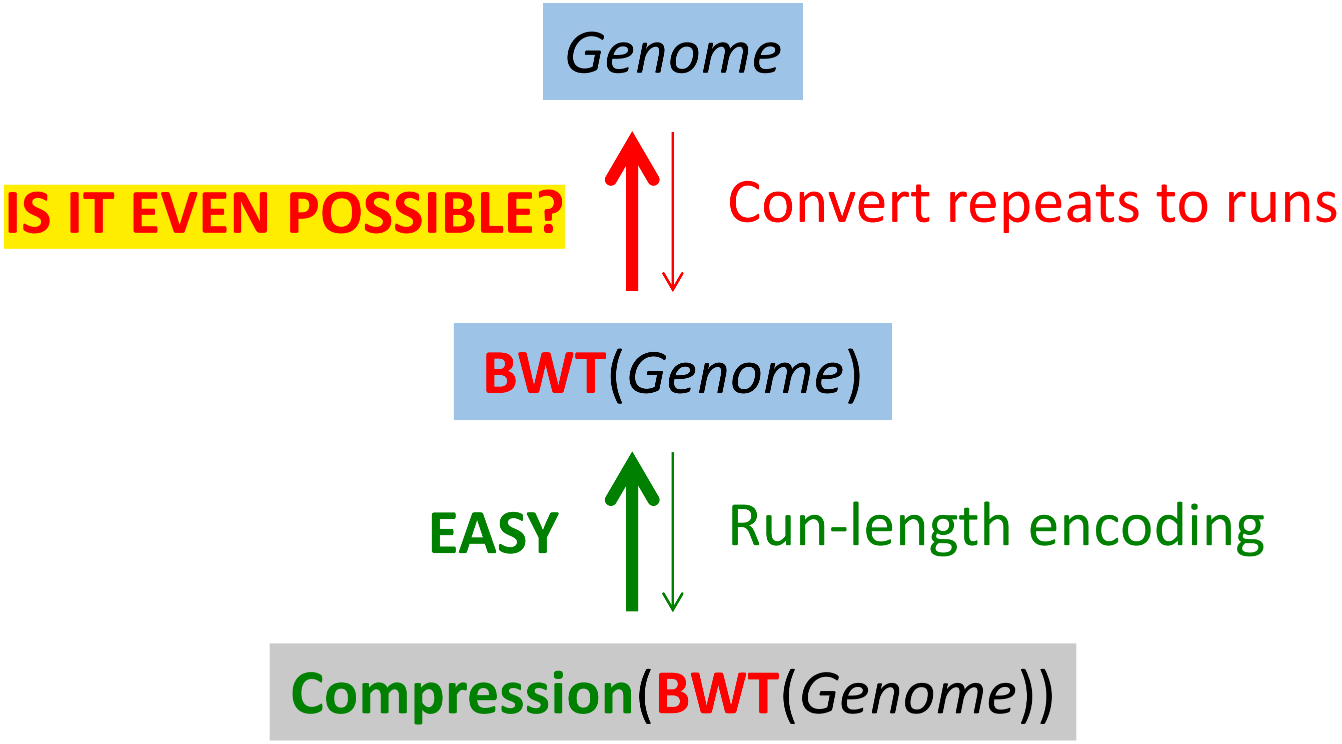
Burrows-Wheeler Transform (BWT):
Last column = **smnpbnnaaaaa$a**

Applying BWT to the Double Helix Paper by Watson&Crick

```
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
```

"and" is a frequent repeat in English texts

Going Back From BWT(*Genome*) to *Genome*



Genome

IS IT EVEN POSSIBLE?

Convert repeats to runs

BWT(*Genome*)

EASY

Run-length encoding

Compression(**BWT**(*Genome*))

Outline

- Burrows-Wheeler Transform
- **Inverting Burrows-Wheeler Transform**
- Using BWT for Pattern Matching
- Suffix Arrays
- Approximate Pattern Matching

Reconstructing `banana` from `annb$aa`

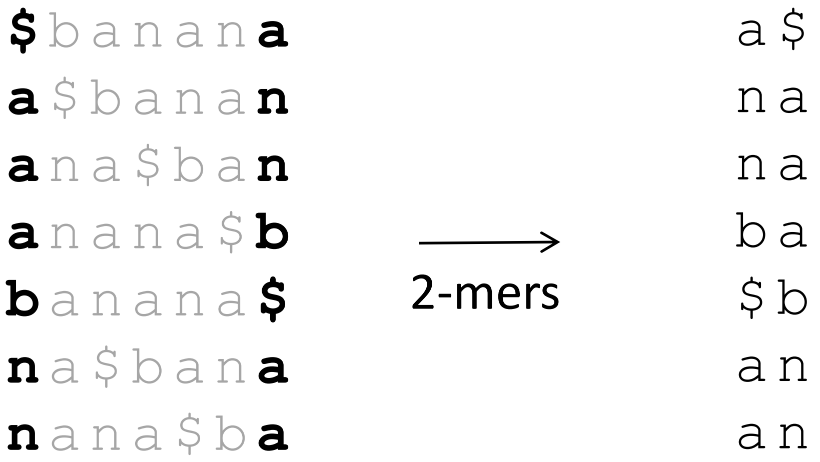
```
$banana
a$banan
ana$ban
anana$b
banana$
na$bana
nana$ba
```

Reconstructing `banana`

```
$banana
a$banan
ana$ban
anana$b
banana$
na$bana
nana$ba
```

- Sorting all elements of “annb$aa” gives first column of BWT matrix.

Reconstructing banana



$banan**a**

a$bana**n**

ana$ba**n**

anana$**b**

banana**$**

na$ban**a**

nana$b**a**

2-mers

a$

na

na

ba

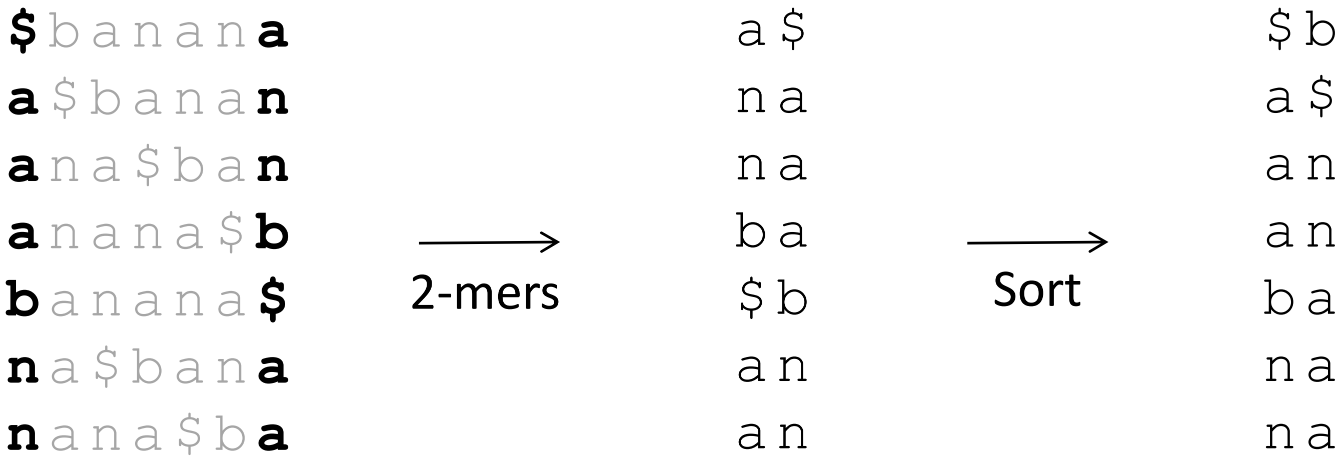
$b

an

an

- We now know 2-mer composition of the circular string banana$

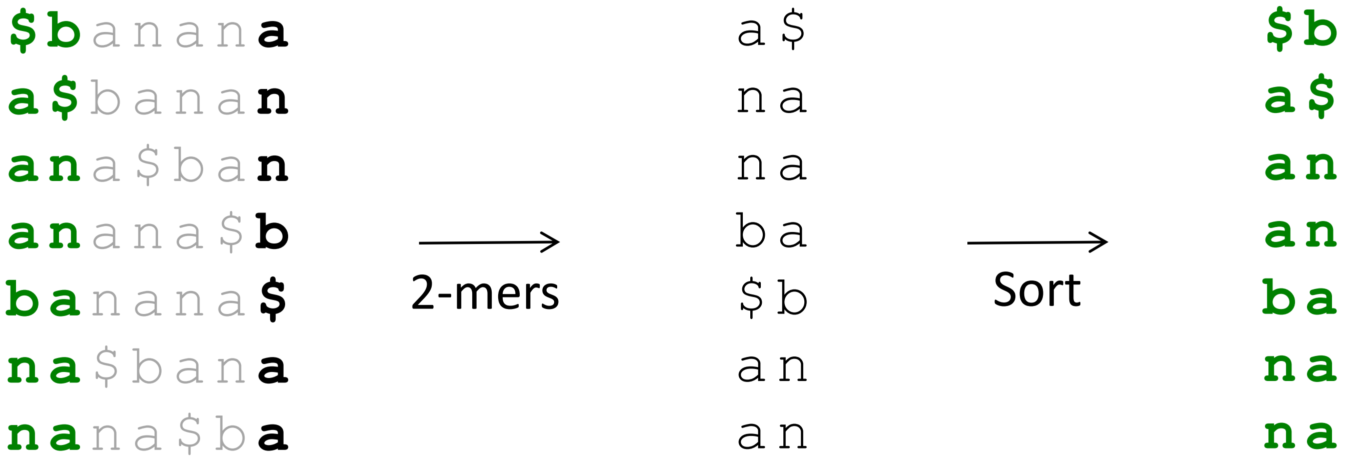
Reconstructing `banana`



$banan**a**		a$		$b
a$bana**n**		na		a$
ana$ba**n**		na		an
anana$**b**	→	ba	→	an
banana**$**	2-mers	$b	Sort	ba
na$ban**a**		an		na
nana$b**a**		an		na

- We now know 2-mer composition of the circular string `banana$`
- Sorting gives us the first 2 columns of the matrix.

Reconstructing banana



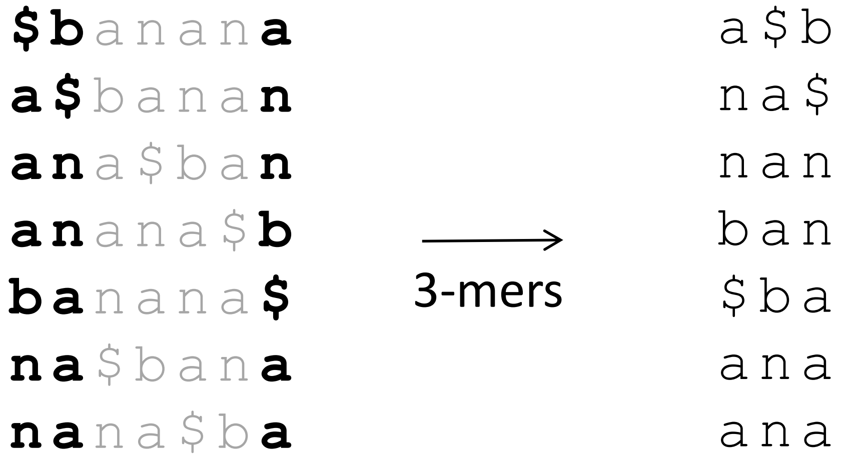
$banana		a$		$b
a$banan		na		a$
ana$ban		na		an
anana$b	→ 2-mers	ba	→ Sort	an
banana$		$b		ba
na$bana		an		na
nana$ba		an		na

- We now know 2-mer composition of the circular string banana$
- Sorting gives us the first 2 columns of the matrix.

Reconstructing banana

```
$banana
a$banan
ana$ban
anana$b
banana$
na$bana
nana$ba
```

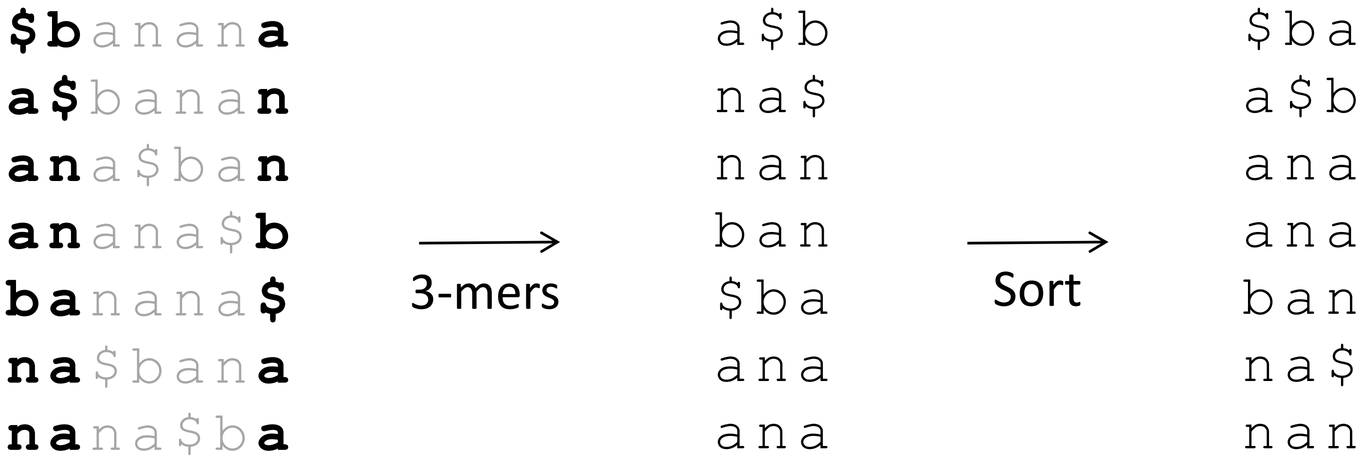
Reconstructing banana



```
$banana            a$b
a$banan            na$
ana$ban            nan
anana$b  ------>   ban
banana$  3-mers    $ba
na$bana            ana
nana$ba            ana
```

- We now know 3-mer composition of the circular string banana$

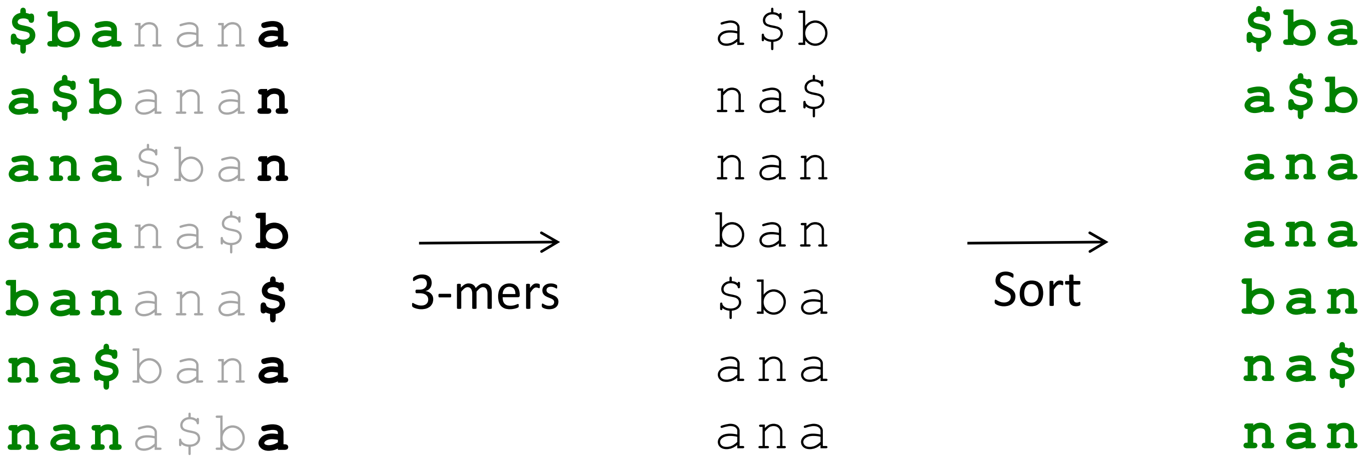
Reconstructing banana



```
$banana                 a$b                $ba
a$banan                 na$                a$b
ana$ban                 nan                ana
anana$b   ———————→      ban   ———————→     ana
banana$    3-mers       $ba     Sort       ban
na$bana                 ana                na$
nana$ba                 ana                nan
```

- We now know 3-mer composition of the circular string banana$
- Sorting gives us the first 3 columns of the matrix.

Reconstructing banana



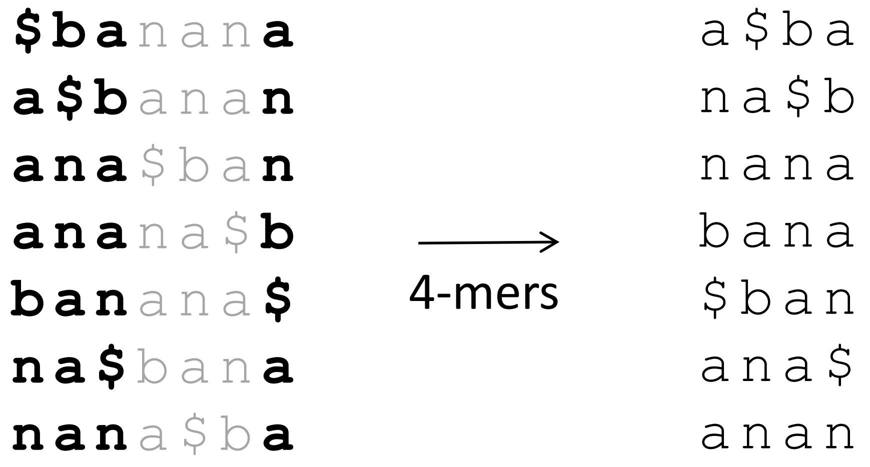
```
$banana                a$b               $ba
a$banan                na$               a$b
ana$ban                nan               ana
anana$b   ------>      ban   ------>     ana
banana$   3-mers       $ba   Sort        ban
na$bana                ana               na$
nana$ba                ana               nan
```

- We now know 3-mer composition of the circular string banana$
- Sorting gives us the first 3 columns of the matrix.

Reconstructing banana

```
$banana
a$banan
ana$ban
anana$b
banana$
na$bana
nana$ba
```

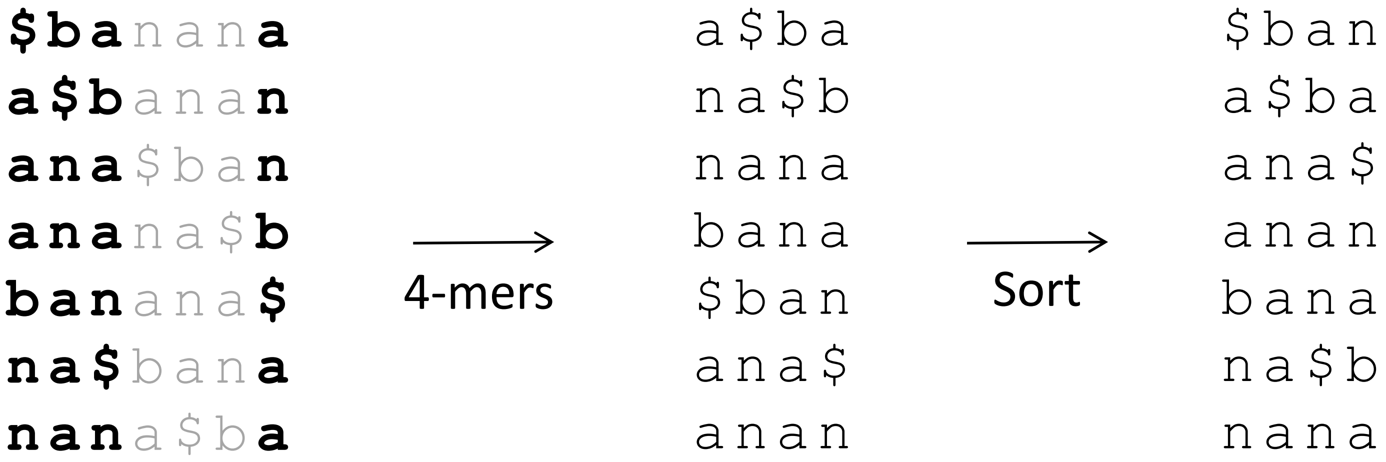
Reconstructing banana



```
$banana        a$ba
a$banan        na$b
ana$ban        nana
anana$b  --->  bana
banana$ 4-mers $ban
na$bana        ana$
nana$ba        anan
```

- We now know 4-mer composition of the circular string banana$

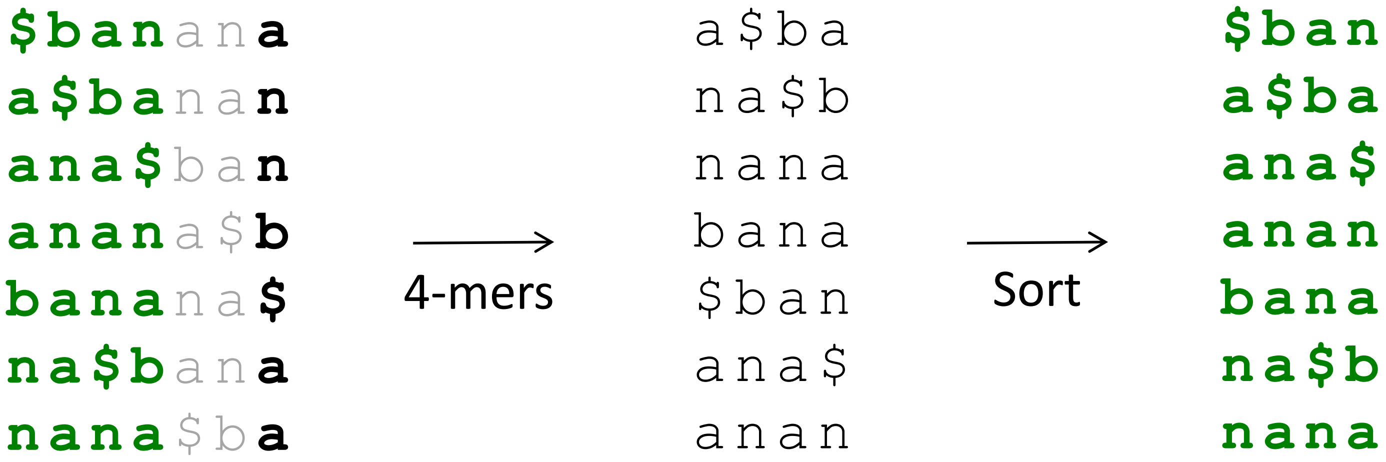
Reconstructing banana



```
$banana          a$ba          $ban
a$banan          na$b          a$ba
ana$ban          nana          ana$
anana$b  ---->   bana  ---->   anan
banana$  4-mers  $ban  Sort    bana
na$bana          ana$          na$b
nana$ba          anan          nana
```

- We now know 4-mer composition of the circular string banana$
- Sorting gives us the first 4 columns of the matrix.

Reconstructing banana



$banana
a$banan
ana$ban
anana$b
banana$
na$bana
nana$ba

4-mers

a$ba
na$b
nana
bana
$ban
ana$
anan

Sort

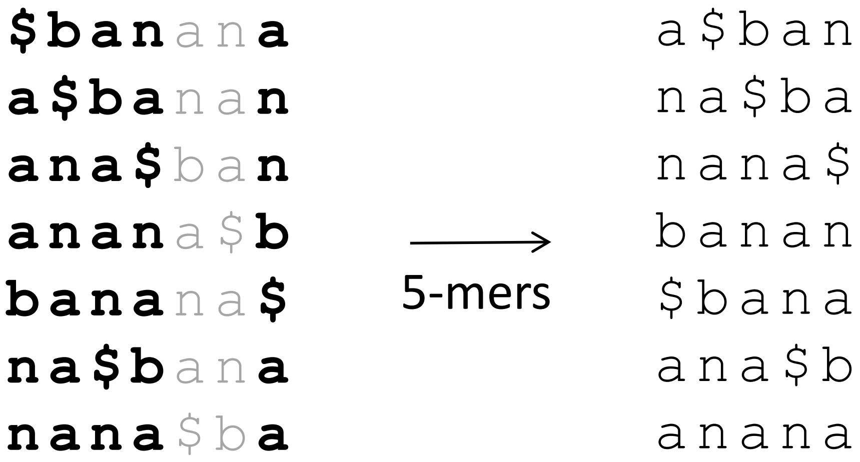
$ban
a$ba
ana$
anan
bana
na$b
nana

- We now know 4-mer composition of the circular string banana$
- Sorting gives us the first 4 columns of the matrix.

Reconstructing banana

```
$banana
a$banan
ana$ban
anana$b
banana$
na$bana
nana$ba
```

Reconstructing banana



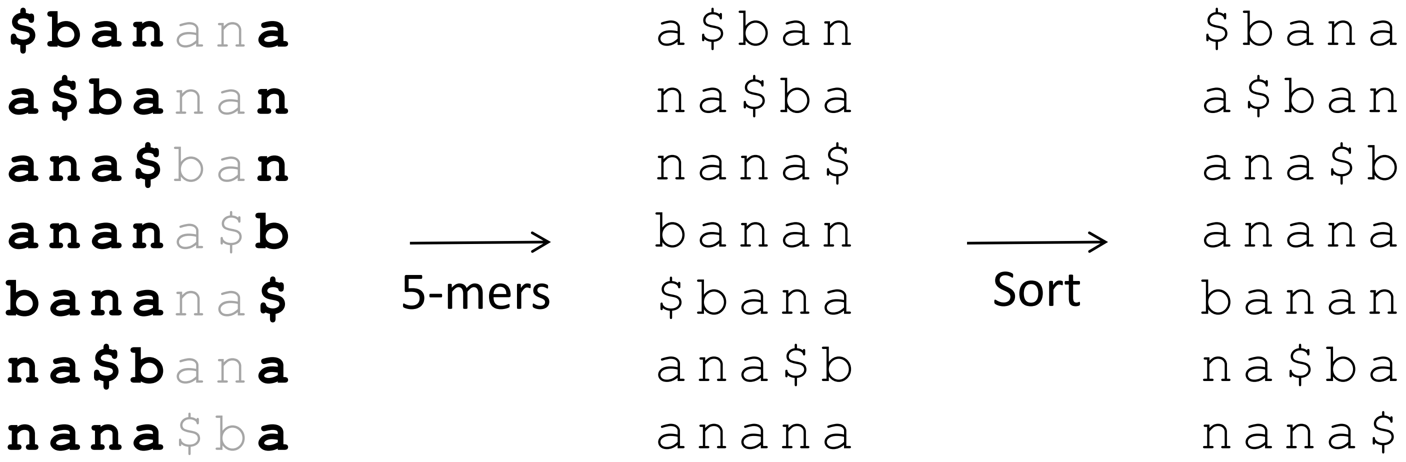
$banan**a**
a$bana**n**
ana$ba**n**
anana$**b**
banana**$**
na$ban**a**
nana$b**a**

5-mers

a$ban
na$ba
nana$
banan
$bana
ana$b
anana

- We now know 5-mer composition of the circular string banana$

Reconstructing `banana`



$banana**a**
a$bana**n**
ana$ba**n**
anana$**b**
banana**$**
na$ban**a**
nana$b**a**

5-mers

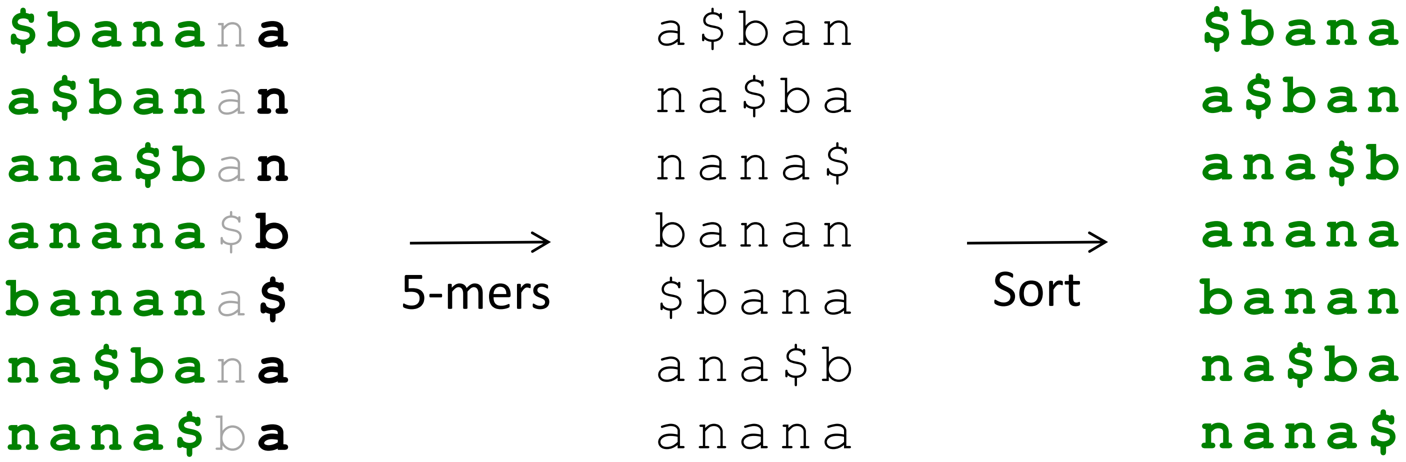
a$ban
na$ba
nana$
banan
$bana
ana$b
anana

Sort

$bana
a$ban
ana$b
anana
banan
na$ba
nana$

- We now know 5-mer composition of the circular string `banana$`
- Sorting gives us the first 5 columns of the matrix.

Reconstructing banana



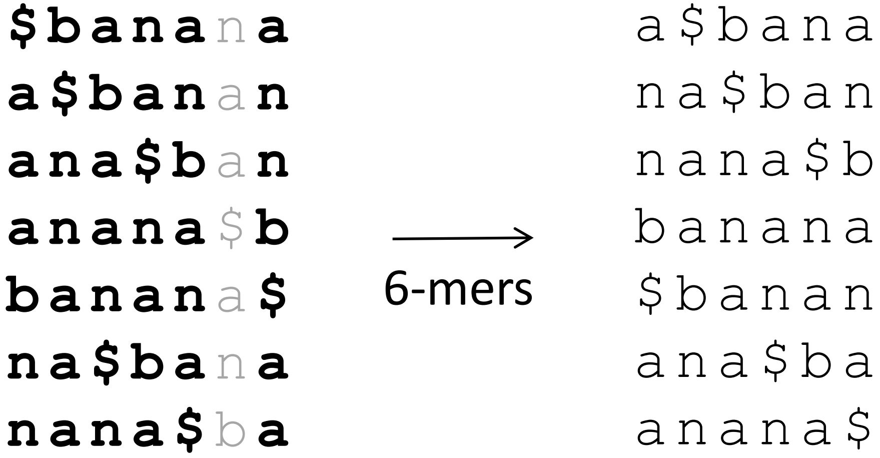
```
$banana           a$ban          $bana
a$banan           na$ba          a$ban
ana$ban           nana$          ana$b
anana$b  ──────→  banan  ─────→  anana
banana$  5-mers   $bana  Sort    banan
na$bana           ana$b          na$ba
nana$ba           anana          nana$
```

- We now know 5-mer composition of the circular string banana$
- Sorting gives us the first 5 columns of the matrix.

Reconstructing `banana`

```
$banana
a$banan
ana$ban
anana$b
banana$
na$bana
nana$ba
```

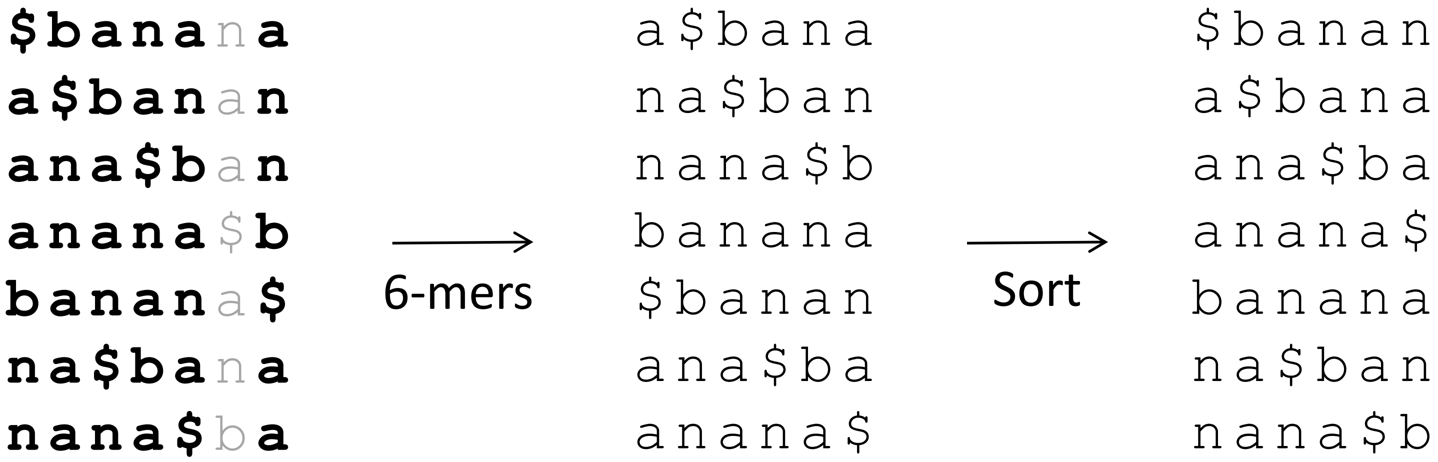
Reconstructing banana



```
$banana        a$bana
a$banan        na$ban
ana$ban        nana$b
anana$b  -->   banana
banana$ 6-mers $banan
na$bana        ana$ba
nana$ba        anana$
```

- We now know 6-mer composition of the circular string banana$

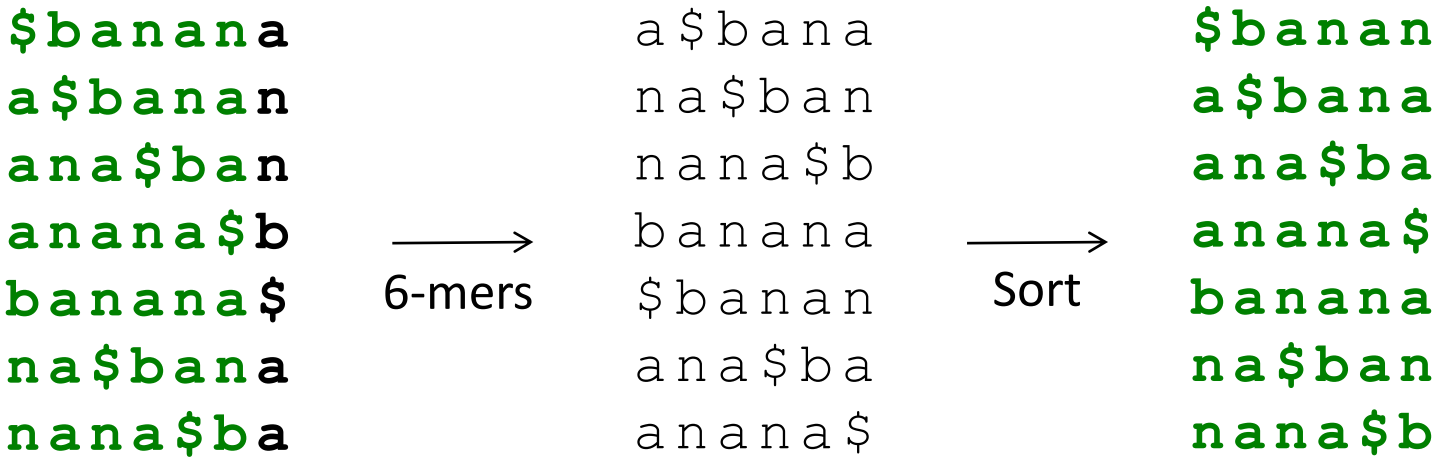
Reconstructing banana



```
$banana              a$bana            $banan
a$banan              na$ban            a$bana
ana$ban              nana$b            ana$ba
anana$b   ------->   banana  ------->  anana$
banana$   6-mers     $banan    Sort    banana
na$bana              ana$ba            na$ban
nana$ba              anana$            nana$b
```

- We now know 6-mer composition of the circular string banana$
- Sorting gives us the first 6 columns of the matrix.

Reconstructing banana



```
$banana
a$banan
ana$ban
anana$b
banana$
na$bana
nana$ba
```

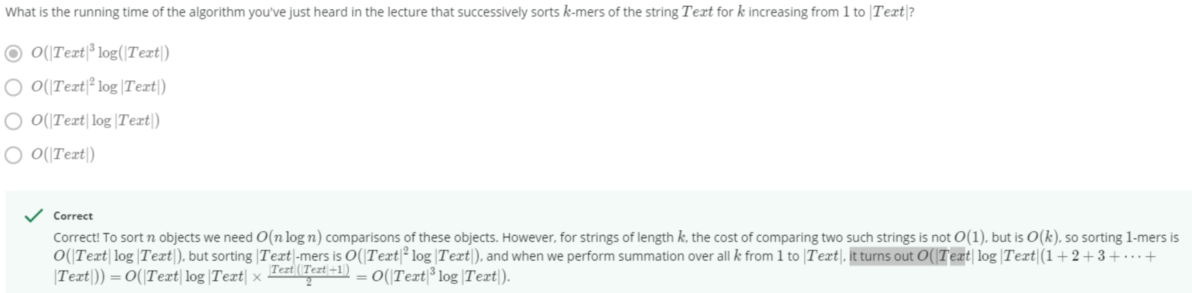
6-mers

```
a$bana
na$ban
nana$b
banana
$banan
ana$ba
anana$
```

Sort

```
$banan
a$bana
ana$ba
anana$
banana
na$ban
nana$b
```

- We now know 6-mer composition of the circular string banana$
- Sorting gives us the first 6 columns of the matrix



What is the running time of the algorithm you've just heard in the lecture that successively sorts k-mers of the string $Text$ for k increasing from 1 to $|Text|$?

● $O(|Text|^3 \log(|Text|)$

○ $O(|Text|^2 \log |Text|)$

○ $O(|Text| \log |Text|)$

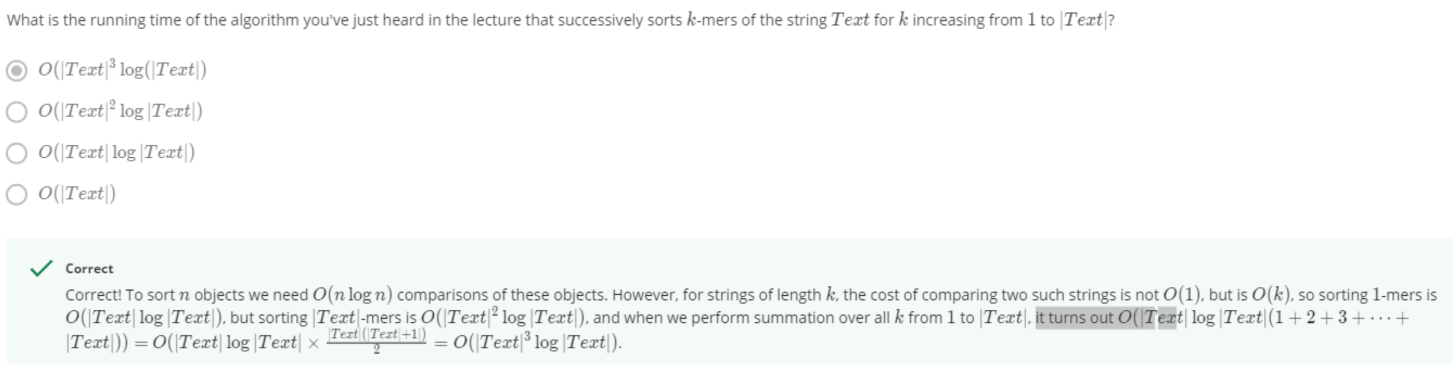
○ $O(|Text|)$

✓ **Correct**

Correct! To sort n objects we need $O(n \log n)$ comparisons of these objects. However, for strings of length k, the cost of comparing two such strings is not $O(1)$, but is $O(k)$, so sorting 1-mers is $O(|Text| \log |Text|)$, but sorting $|Text|$-mers is $O(|Text|^2 \log |Text|)$, and when we perform summation over all k from 1 to $|Text|$, it turns out $O(|Text| \log |Text|(1+2+3+\cdots+|Text|)) = O(|Text| \log |Text| \times \frac{|Text|(|Text|+1|)}{2} = O(|Text|^3 \log |Text|)$.

Reconstructing `banana`

```
$banana
a$banan
ana$ban
anana$b
banana$
na$bana
nana$ba
```



What is the running time of the algorithm you've just heard in the lecture that successively sorts k-mers of the string $Text$ for k increasing from 1 to $|Text|$?

● $O(|Text|^3 \log(|Text|)$

○ $O(|Text|^2 \log |Text|)$

○ $O(|Text| \log |Text|)$

○ $O(|Text|)$

✓ **Correct**

Correct! To sort n objects we need $O(n \log n)$ comparisons of these objects. However, for strings of length k, the cost of comparing two such strings is not $O(1)$, but is $O(k)$, so sorting 1-mers is $O(|Text| \log |Text|)$, but sorting $|Text|$-mers is $O(|Text|^2 \log |Text|)$, and when we perform summation over all k from 1 to $|Text|$, it turns out $O(|Text| \log |Text|(1+2+3+\cdots+|Text|)) = O(|Text| \log |Text| \times \frac{|Text|(|Text|+1|)}{2} = O(|Text|^3 \log |Text|)$.

- We now know the entire matrix!

Reconstructing banana

```
$banana
a$banan
ana$ban
anana$b
banana$
na$bana
nana$ba
```

- We now know the entire matrix!
- Symbols in the first row (after $) spell **banana**.

More Memory Issues

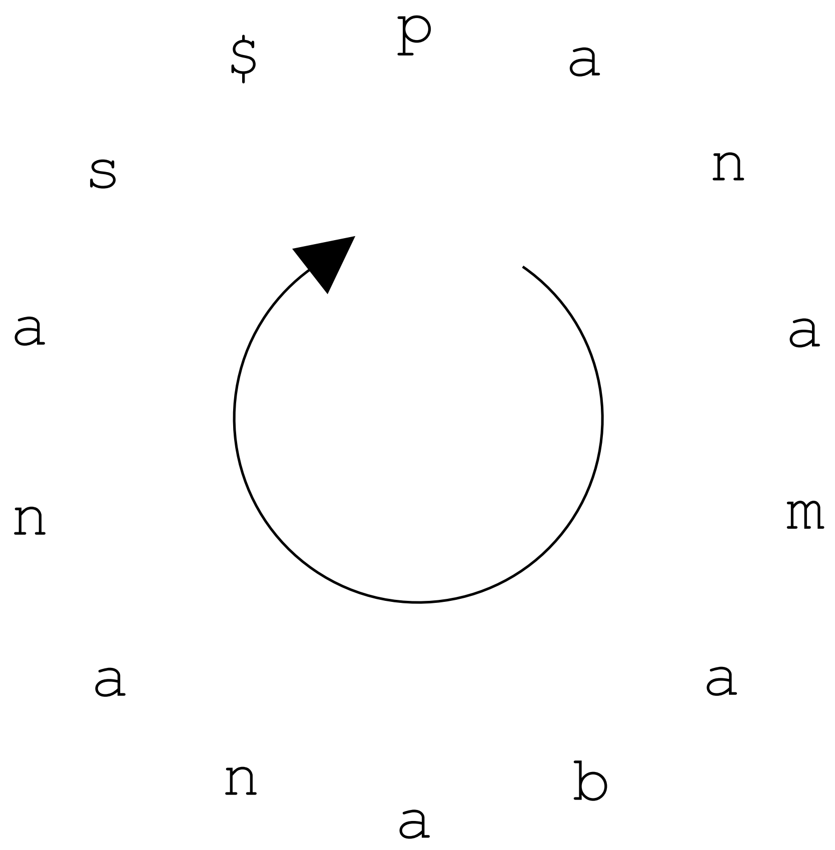
- Reconstructing *Text* from *BWT*(*Text*) required us to store |*Text*| cyclic rotations of |*Text*|.

```
$banana
a$banan
ana$ban
anana$b
banana$
na$bana
nana$ba
```

- Can we invert BWT(*Text*) with less space and without |*Text*| rounds of sorting?

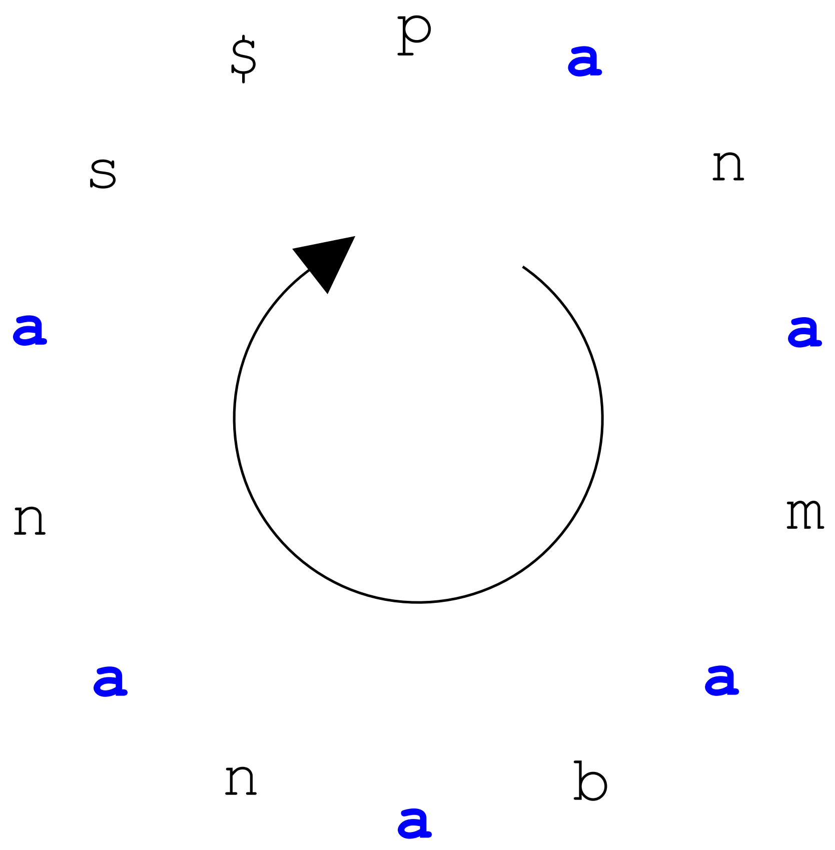
A Strange Observation

```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```



A Strange Observation

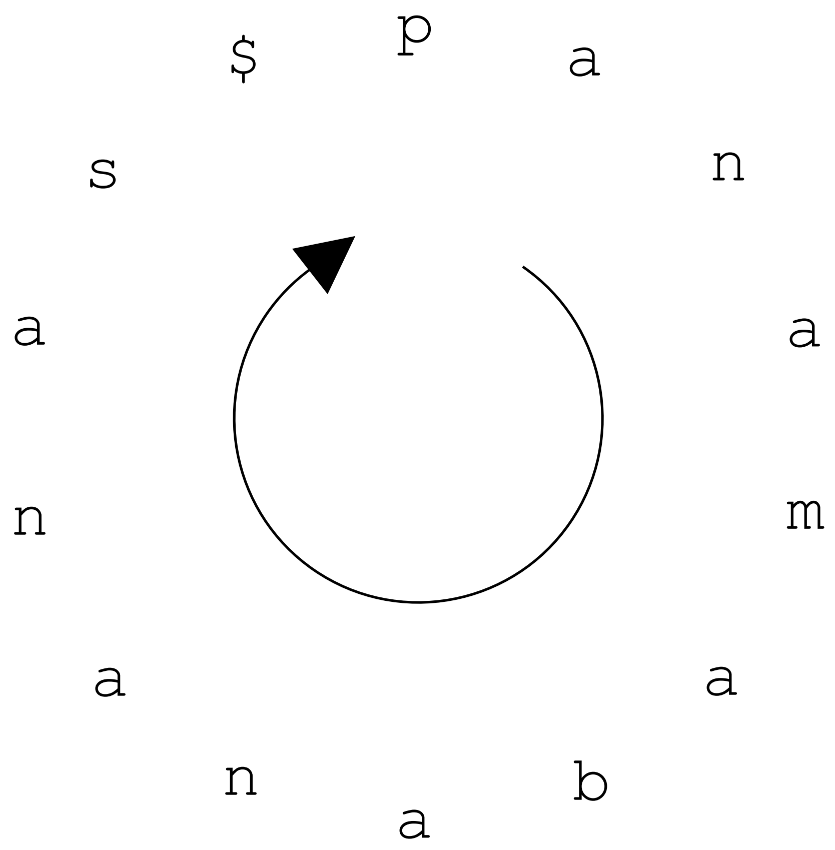
```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```



A Strange Observation

Where is first “a” hiding inside the circle?

```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```

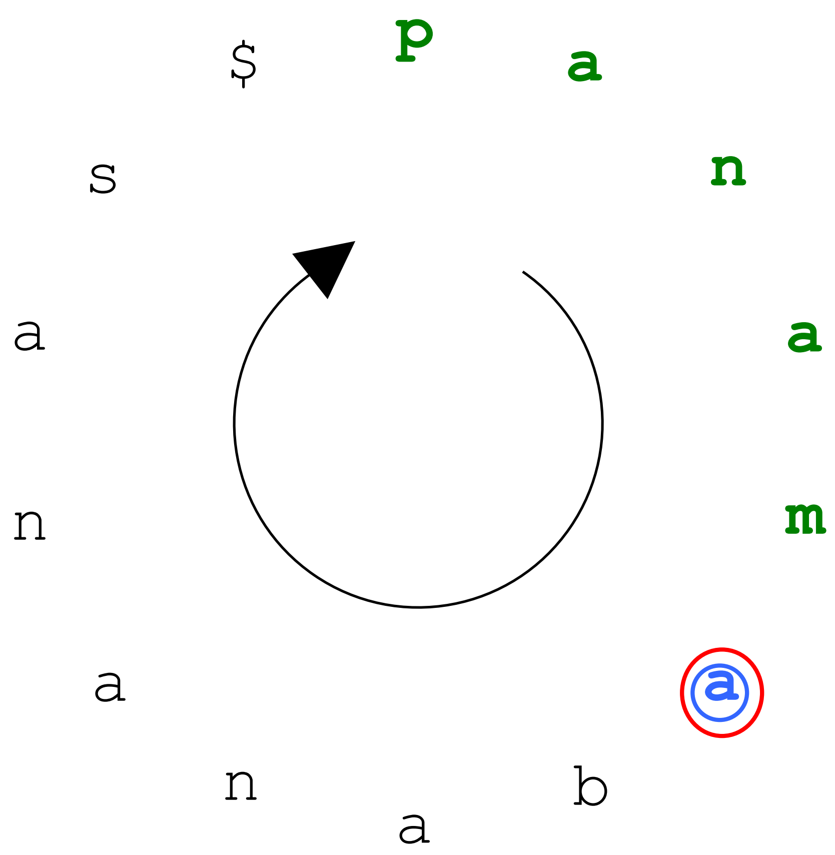


A Strange Observation

Where is first “a” hiding inside the circle?

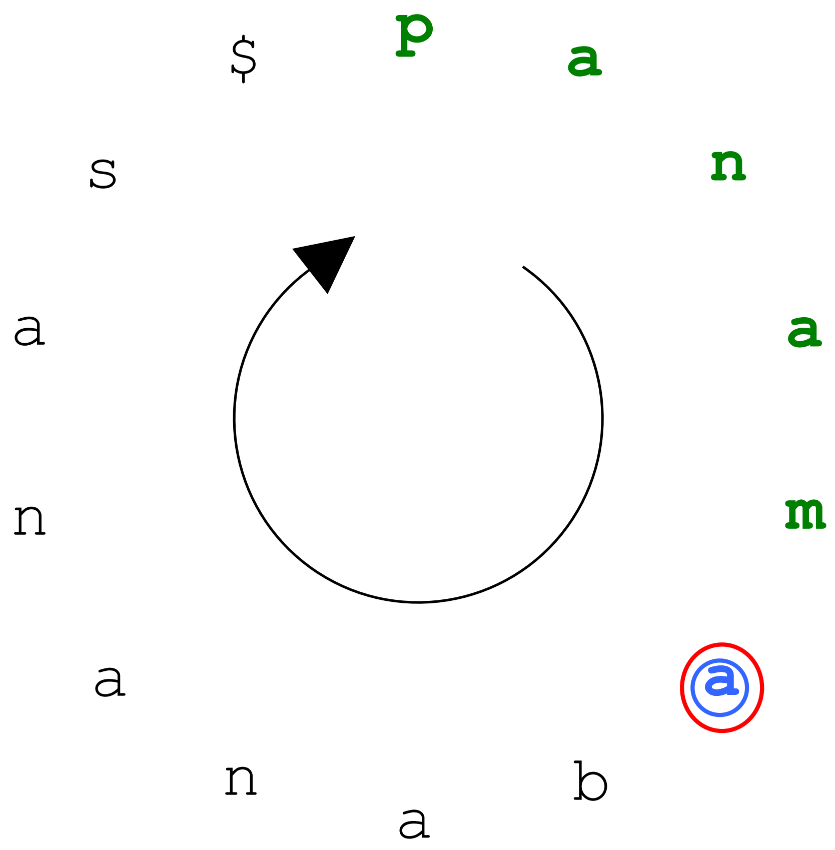
```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```

Where is first “a” hiding inside the circle?



They Are Hiding at the Same Position!

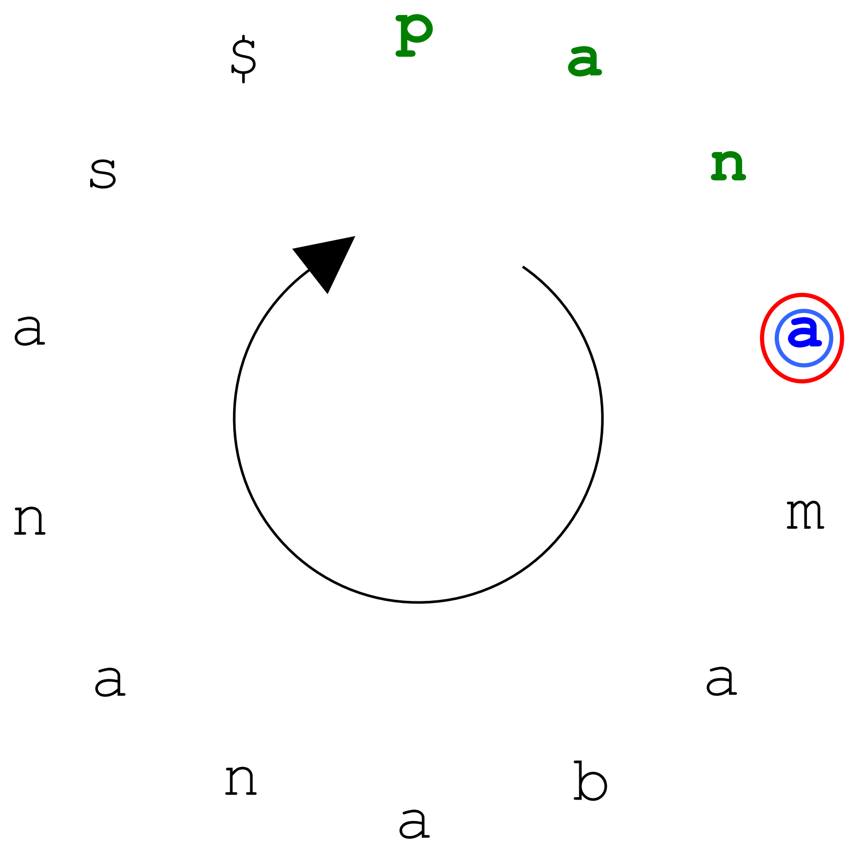
```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```



1st **a** in *FirstColumn* and 1st **a** in *LastColumn*
are hiding at the same position along the cycle!

Another Strange Observation

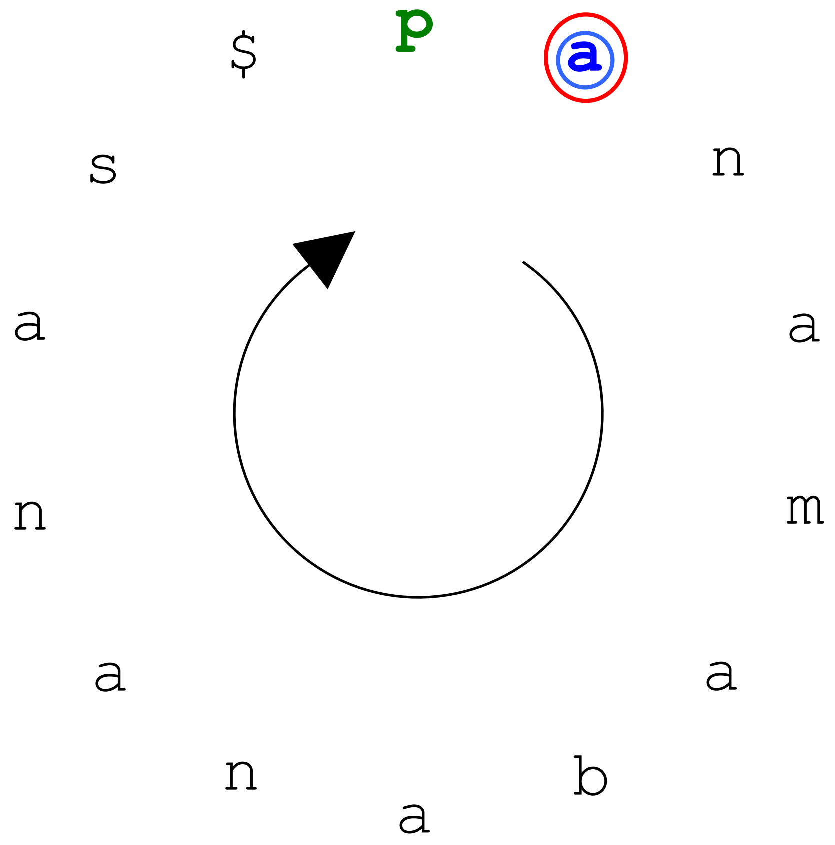
```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```



2nd **a** in *FirstColumn* and 2nd **a** in *LastColumn*
are hiding at the same position along the cycle!

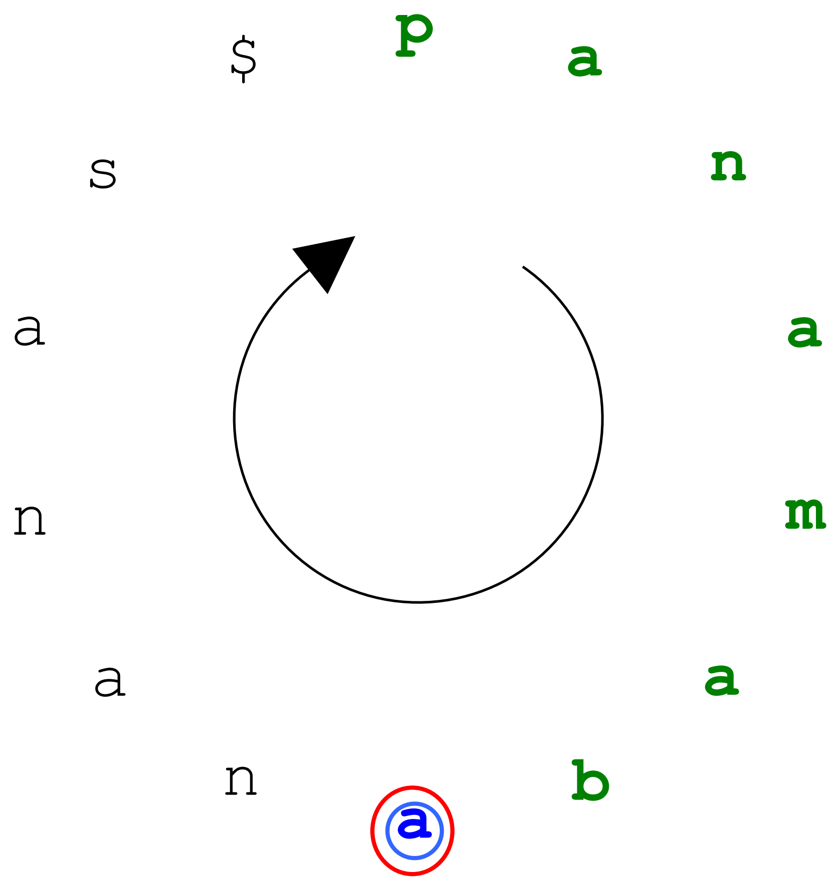
Another Strange Observation

```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```



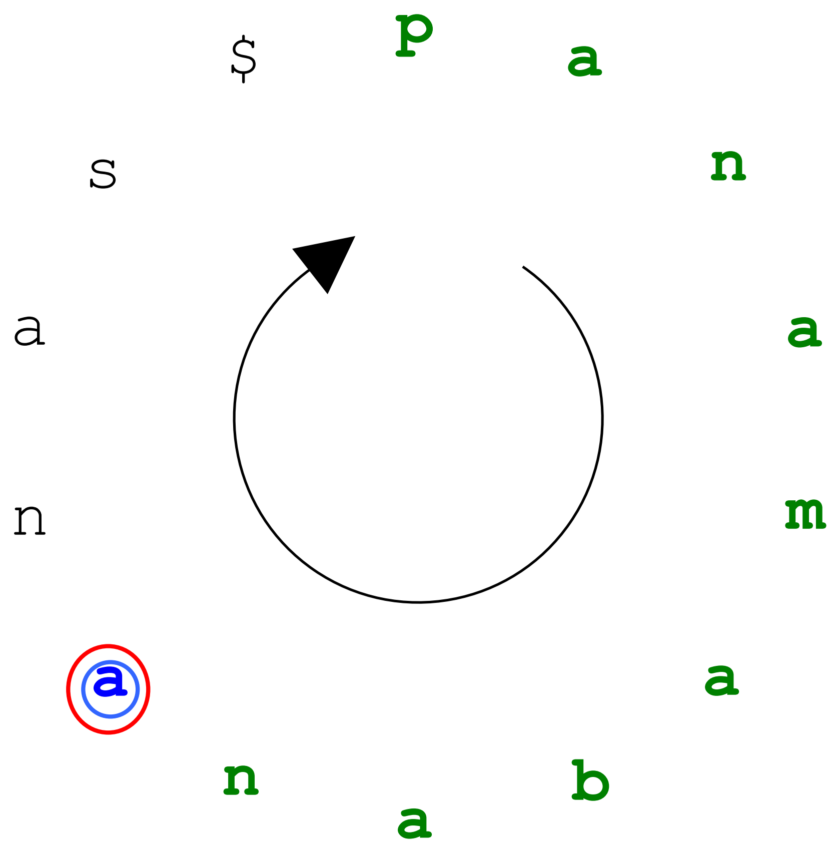
Another Strange Observation

```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```



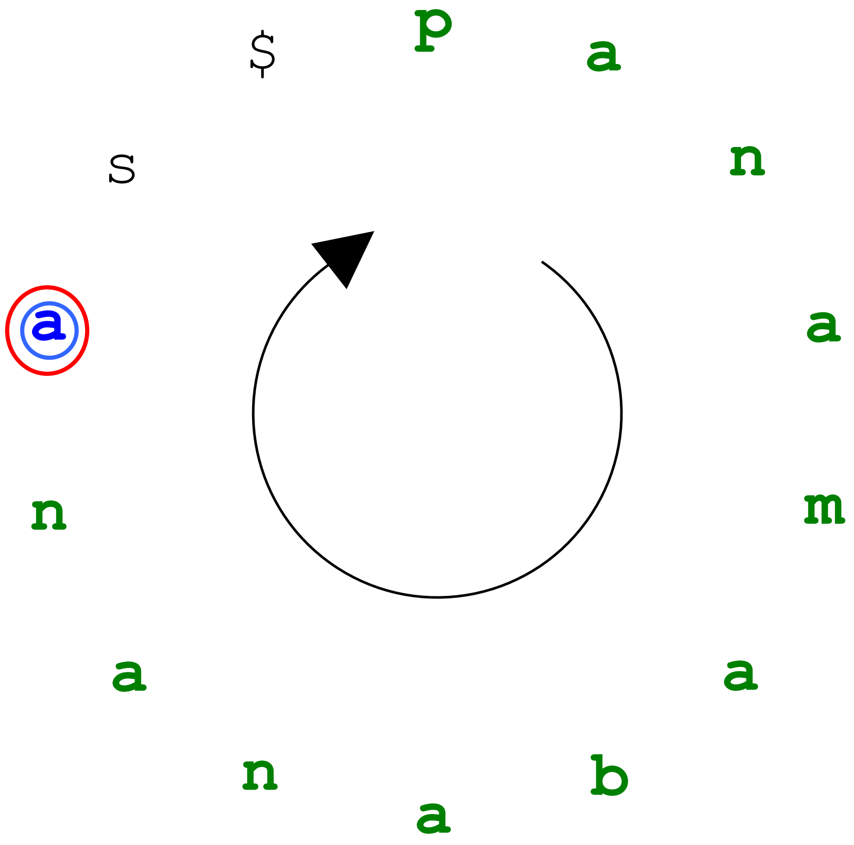
Another Strange Observation

```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```



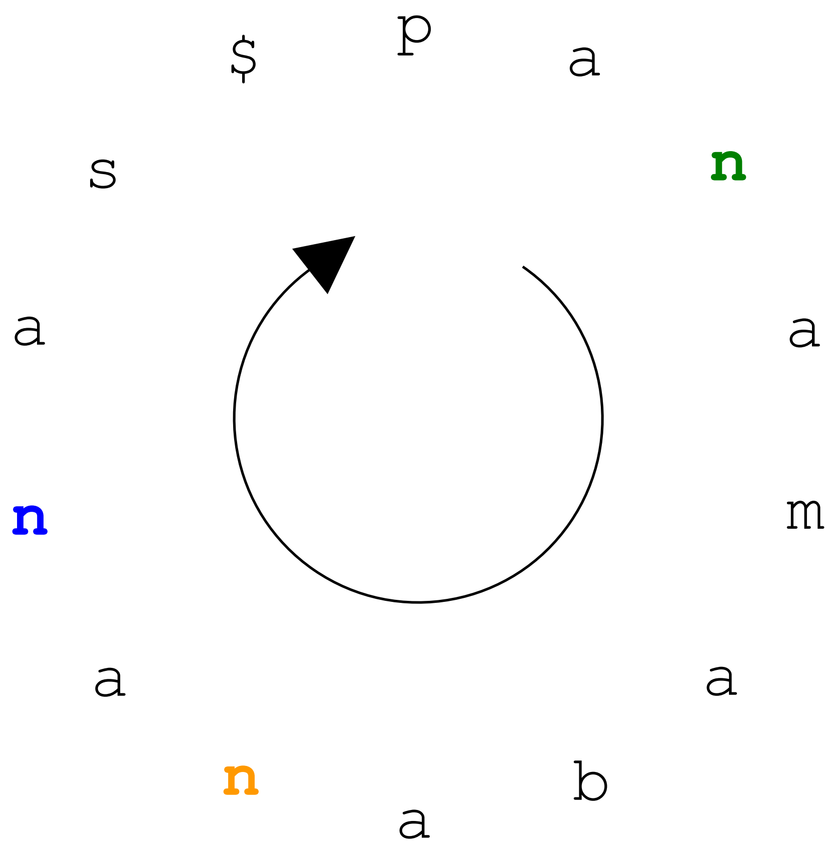
Another Strange Observation

```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panamaban
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```



Another Strange Observation

```
$panamabananas
abananas$panam
amabananas$pan
anamabananas$p
ananas$panamab
anas$panaman
as$panamabanan
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
panamabananas$
s$panamabanana
```



Is It True in General?

```
  $panamabananas
1 abananas$panam
2 amabananas$pan
3 anamabananas$p
4 ananas$panamab
5 anas$panamaban
6 as$panamabanan
  bananas$panama
  mabananas$pana
  namabananas$pa
  nanas$panamaba
  nas$panamabana
  panamabananas$
  s$panamabanana
```

These strings are sorted

Is It True in General?

```
  $panamabananas
1 abananas$panam
2 amabananas$pan
3 anamabananas$p
4 ananas$panamab
5 anas$panamaban
6 as$panamabanan
  bananas$panama
  mabananas$pana
  namabananas$pa
  nanas$panamaba
  nas$panamabana
  panamabananas$
  s$panamabanana
```

These strings are sorted

Chop off **a**

```
bananas$panam
mabananas$pan
namabananas$p
nanas$panamab
nas$panamaban
s$panamabanan
```

Is It True in General?

```
  $panamabananas
1 abananas$panam
2 amabananas$pan
3 anamabananas$p
4 ananas$panamab
5 anas$panamaban
6 as$panamabanan
  bananas$panama
  mabananas$pana
  namabananas$pa
  nanas$panamaba
  nas$panamabana
  panamabananas$
  s$panamabanana
```

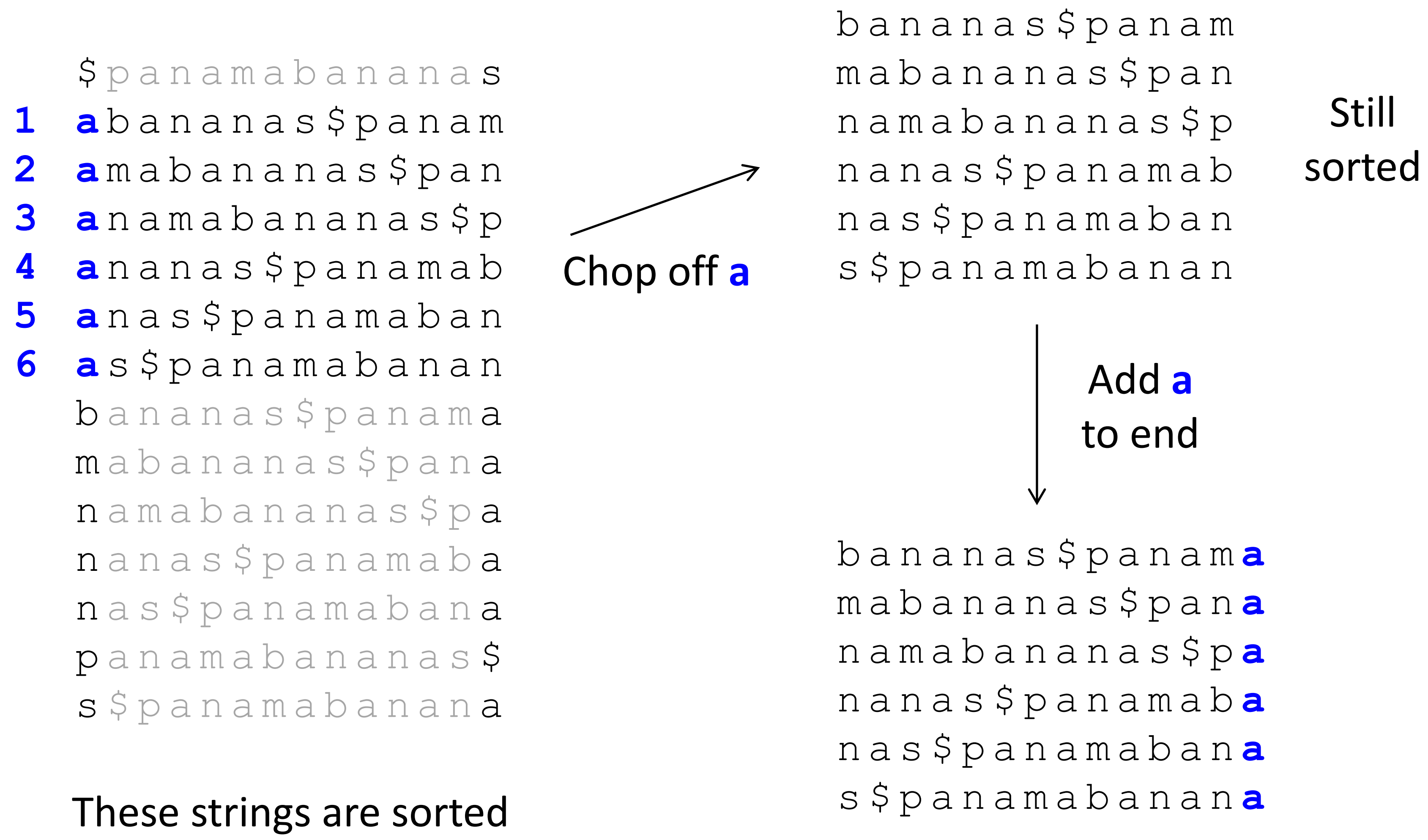
These strings are sorted

Chop off **a**

```
bananas$panam
mabananas$pan
namabananas$p
nanas$panamab
nas$panamaban
s$panamabanan
```

Still sorted

Is It True in General?



```
  $panamabananas
1 abananas$panam
2 amabananas$pan
3 anamabananas$p
4 ananas$panamab
5 anas$panamaban
6 as$panamabanan
  bananas$panama
  mabananas$pana
  namabananas$pa
  nanas$panamaba
  nas$panamabana
  panamabananas$
  s$panamabanana
```

These strings are sorted

Chop off a

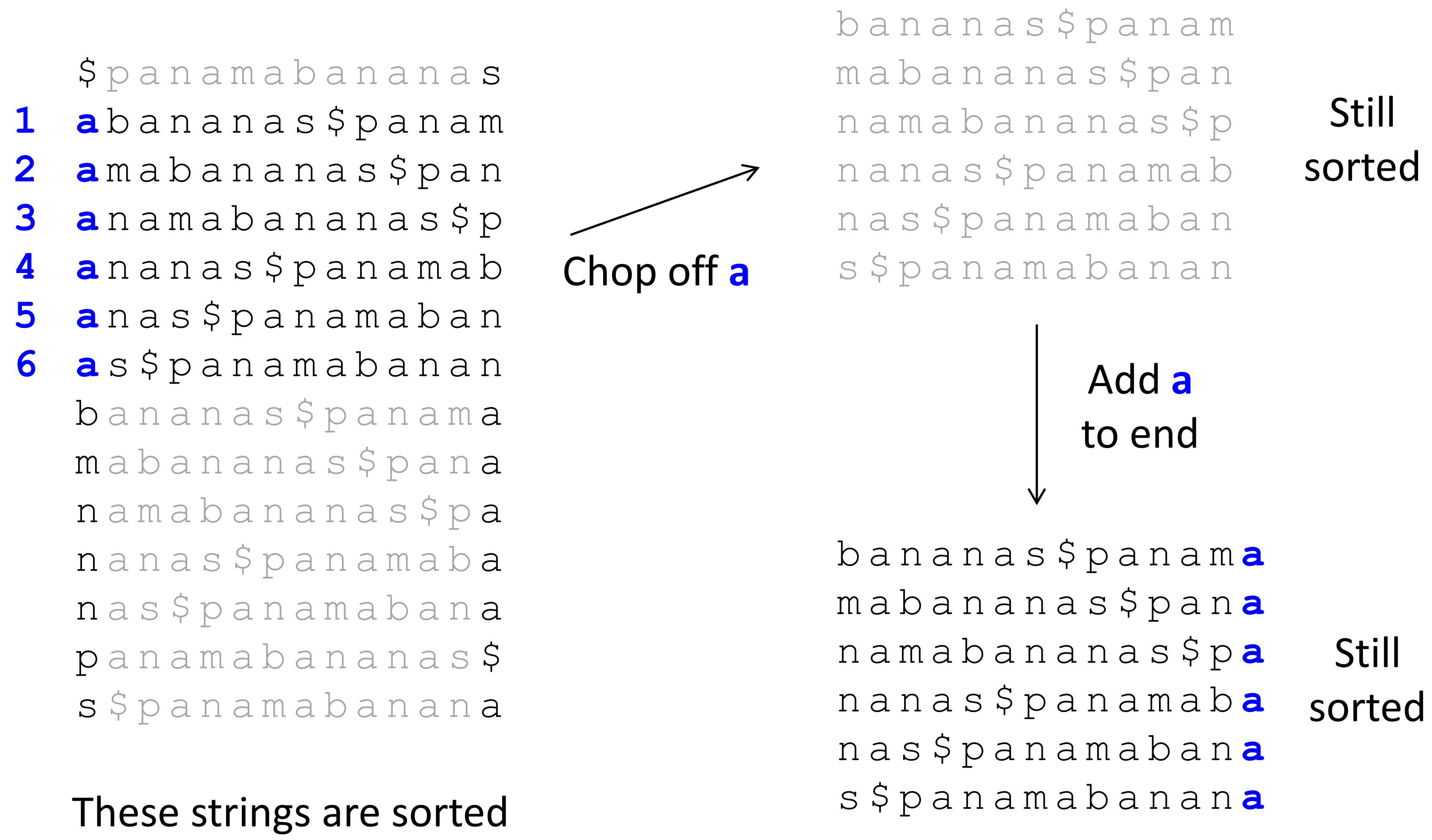
```
bananas$panam
mabananas$pan
namabananas$p
nanas$panamab
nas$panamaban
s$panamabanan
```

Still sorted

Add a to end

```
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
s$panamabanana
```

Is It True in General?



```
  $panamabananas
1 abananas$panam
2 amabananas$pan
3 anamabananas$p
4 ananas$panamab
5 anas$panamaban
6 as$panamabanan
  bananas$panama
  mabananas$pana
  namabananas$pa
  nanas$panamaba
  nas$panamabana
  panamabananas$
  s$panamabanana
```

These strings are sorted

Chop off a

```
bananas$panam
mabananas$pan
namabananas$p
nanas$panamab
nas$panamaban
s$panamabanan
```

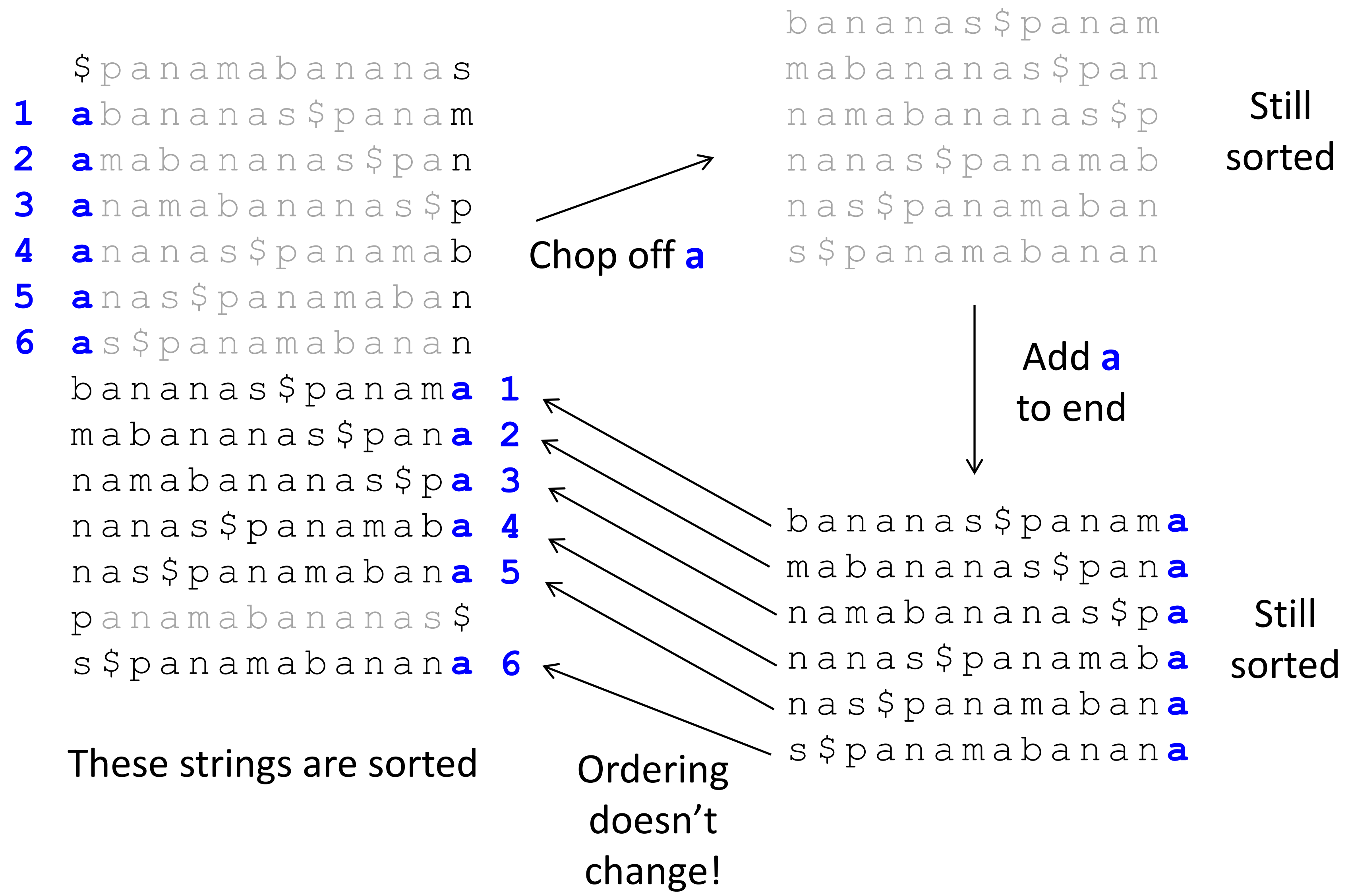
Still sorted

Add a to end

```
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
s$panamabanana
```

Still sorted

Is It True in General?



```
  $panamabananas
1 abananas$panam
2 amabananas$pan
3 anamabananas$p
4 ananas$panamab
5 anas$panamaban
6 as$panamabanan
  bananas$panama 1
  mabananas$pana 2
  namabananas$pa 3
  nanas$panamaba 4
  nas$panamabana 5
  panamabananas$
  s$panamabanana 6
```

These strings are sorted

Chop off a

```
bananas$panam
mabananas$pan
namabananas$p
nanas$panamab
nas$panamaban
s$panamabanan
```

Still sorted

Add a to end

```
bananas$panama
mabananas$pana
namabananas$pa
nanas$panamaba
nas$panamabana
s$panamabanana
```

Still sorted

Ordering doesn't change!

First-Last Property

- the *k*-th occurrence of *symbol* in ***FirstColumn***
- and the *k*-th occurrence of *symbol* in ***LastColumn***
- correspond to appearance of *symbol* at the same position in *Text*.
-

$p_1a_3n_1a_2m_1a_1b_1a_4n_2a_5n_3a_6s_1\$_1$

$\$_1$panamabanana$s_1$
$a_1$bananas\$panam_1
$a_2$mabananas\$pan_1
$a_3$namabananas\$$p_1$
$a_4$nanas\$panamab_1
$a_5$nas\$panamaban_2
a_6s\$panamabanan_3
$b_1$ananas\$panama_1
$m_1$abananas\$pana_2
$n_1$amabananas\$pa_3
$n_2$anas\$panamaba_4
$n_3$as\$panamabana_5
$p_1$anamabananas$\$_1$
s_1\$panamabanana_6

Inverting BWT Again

$\$_1$ panamabanana s_1

a_1 bananas\$pana m_1

a_2 mabananas\$pa n_1

a_3 namabananas\$ p_1

a_4 nanas\$panama b_1

a_5 nas\$panamaba n_2

a_6 s\$panamabana n_3

b_1 ananas\$panam a_1

m_1 abananas\$pan a_2

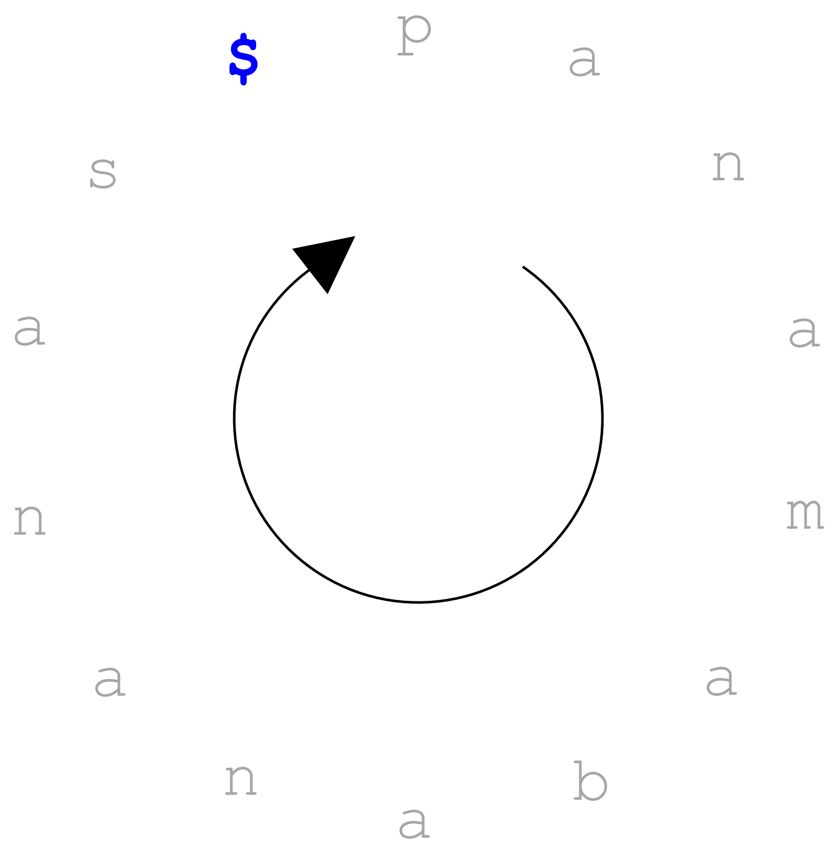
n_1 amabananas\$p a_3

n_2 anas\$panamab a_4

n_3 as\$panamaban a_5

p_1 anamabananas $\$_1$

s_1 \$panamabanan a_6



Inverting BWT Again

$\$_1\text{panamabanana}s_1$

$a_1\text{bananas\$pana}m_1$

$a_2\text{mabananas\$pa}n_1$

$a_3\text{namabananas\$}p_1$

$a_4\text{nanas\$panama}b_1$

$a_5\text{nas\$panamaba}n_2$

$a_6\text{s\$panamabana}n_3$

$b_1\text{ananas\$panam}a_1$

$m_1\text{abananas\$pan}a_2$

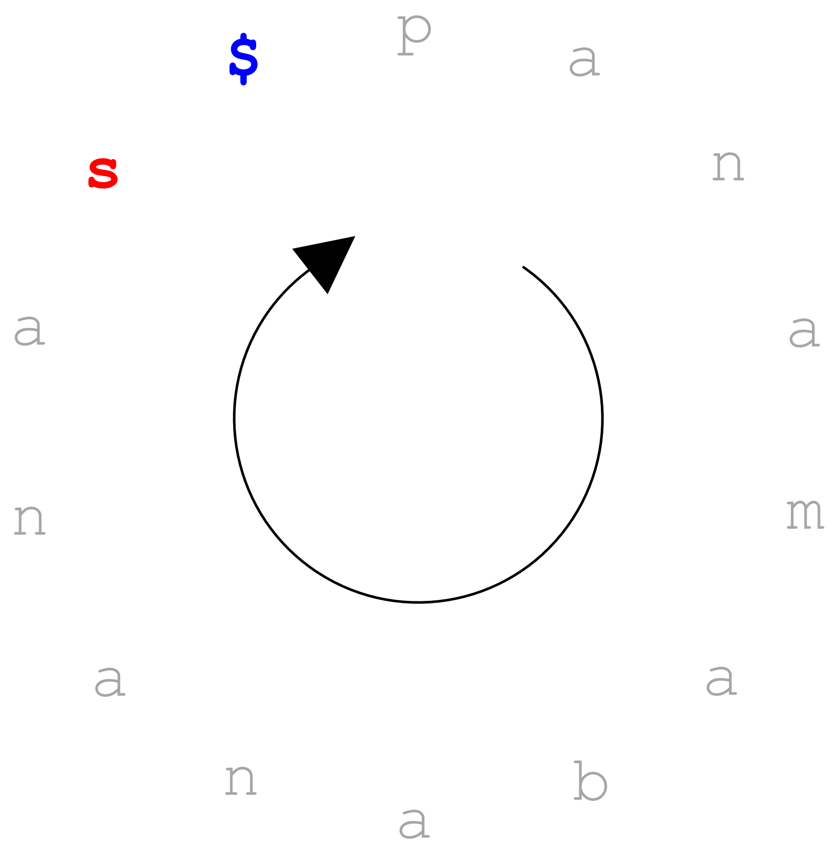
$n_1\text{amabananas\$p}a_3$

$n_2\text{anas\$panamab}a_4$

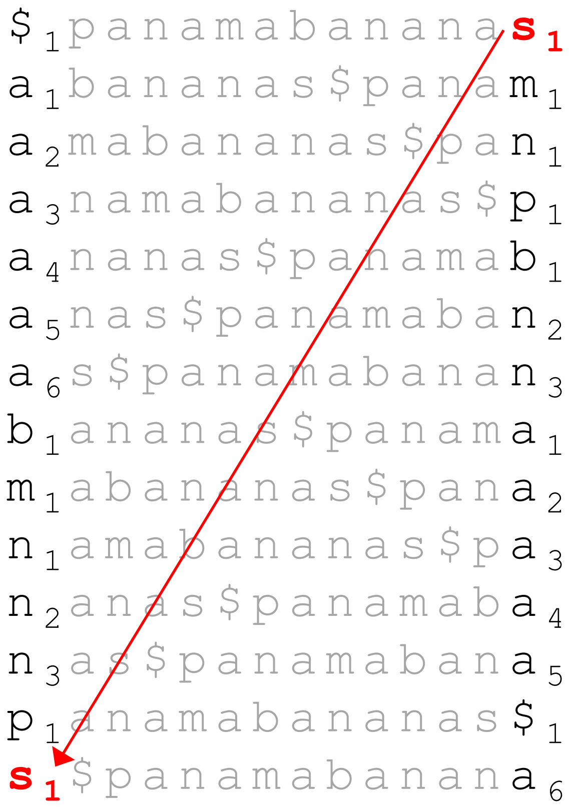
$n_3\text{as\$panamaban}a_5$

$p_1\text{anamabananas}\$_1$

$s_1\text{\$panamabanan}a_6$



Inverting BWT Again



$\$_1$ panamabanana s_1

a_1 bananas$pana m_1

a_2 mabananas$pa n_1

a_3 namabananas$ p_1

a_4 nanas$panama b_1

a_5 nas$panamaba n_2

a_6 s$panamabana n_3

b_1 ananas$panam a_1

m_1 abananas$pan a_2

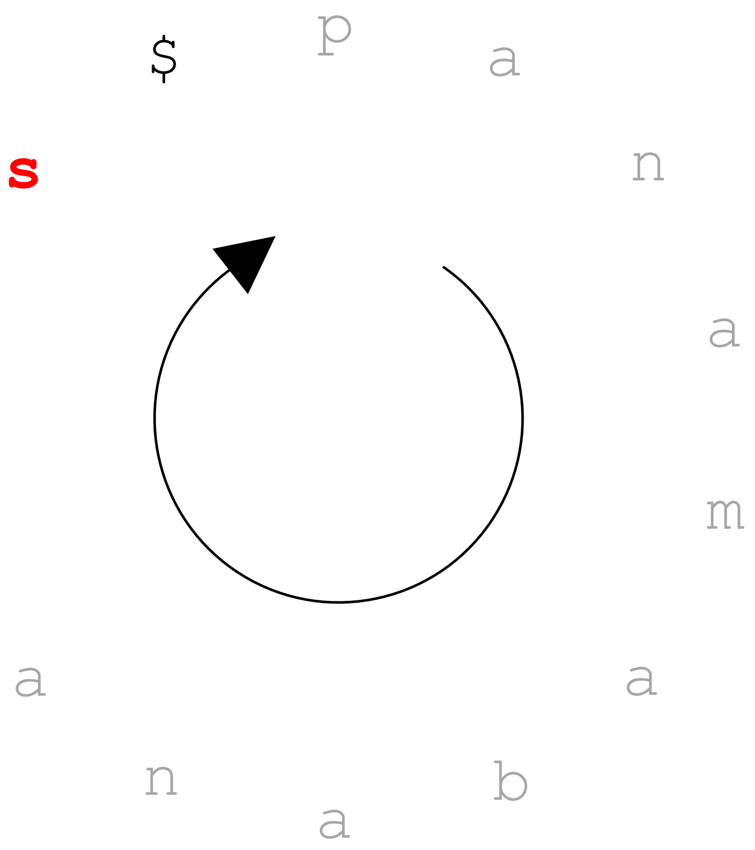
n_1 amabananas$p a_3

n_2 anas$panamab a_4

n_3 as$panamaban a_5

p_1 anamabananas $\$_1$

s_1 $panamabanan a_6



Inverting BWT Again

$\$_1$ panamabanana s_1

a_1 bananas\$pana m_1

a_2 mabananas\$pa n_1

a_3 namabananas\$ p_1

a_4 nanas\$panama b_1

a_5 nas\$panamaba n_2

a_6 s\$panamabana n_3

b_1 ananas\$panam a_1

m_1 abananas\$pan a_2

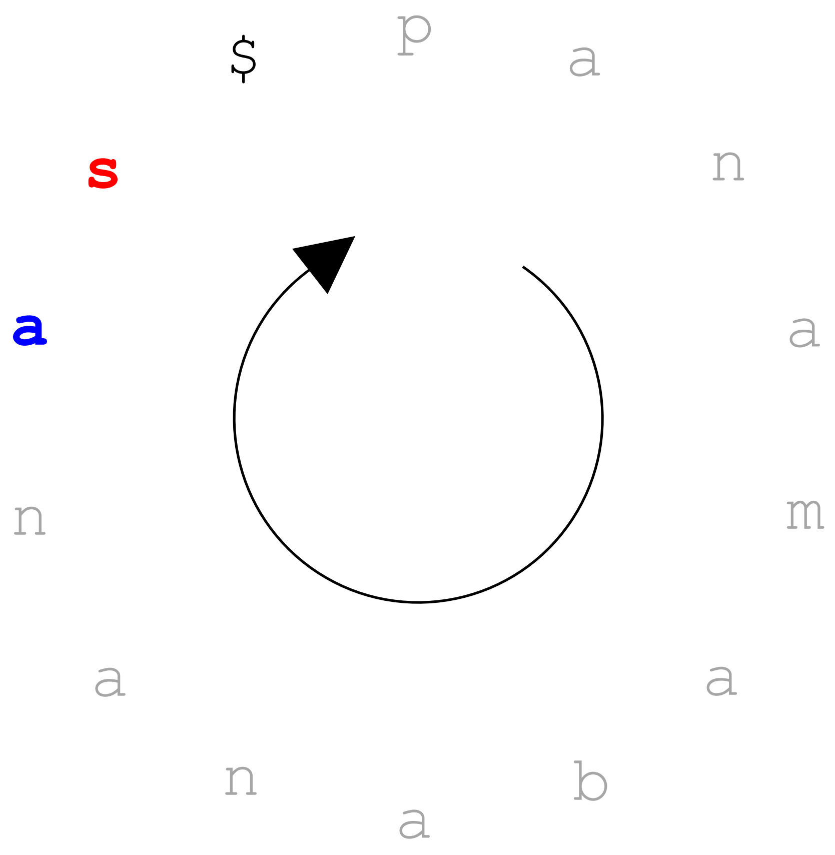
n_1 amabananas\$p a_3

n_2 anas\$panamab a_4

n_3 as\$panamaban a_5

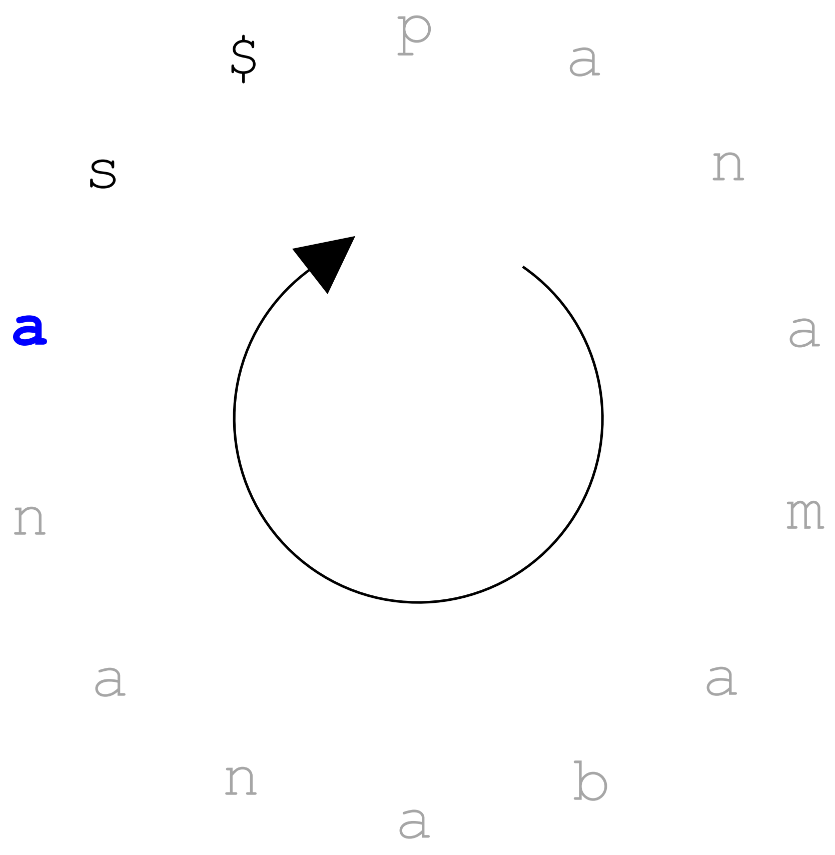
p_1 anamabananas $\$_1$

s_1 \$panamabanan a_6



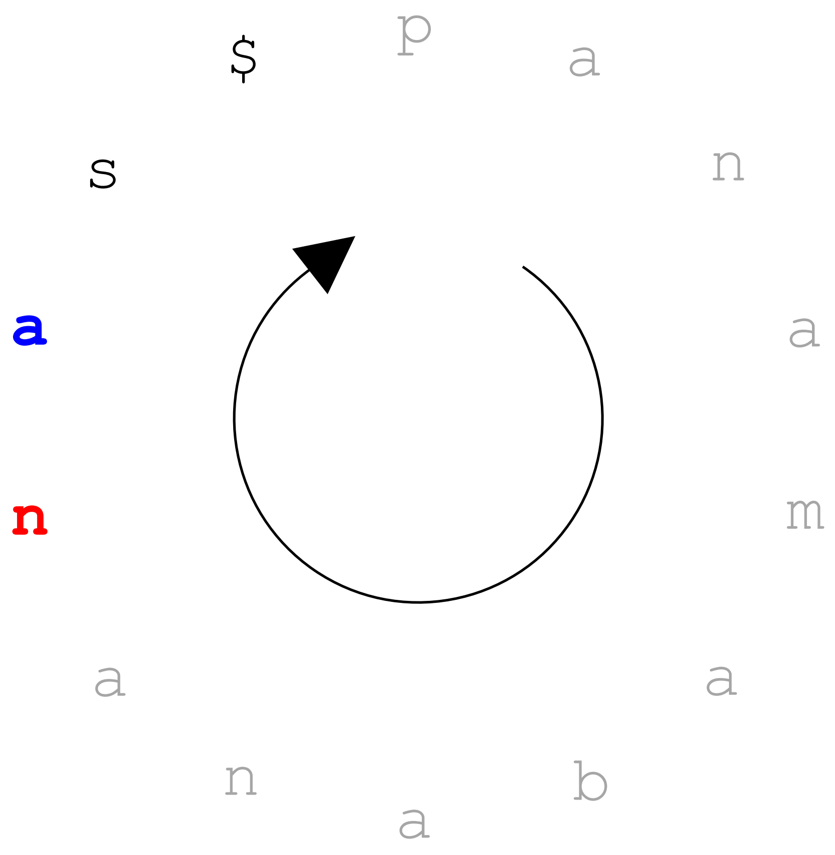
Inverting BWT Again

$\$_1$ panamabanana s_1
a_1 bananas\$pana m_1
a_2 mabananas\$pa n_1
a_3 namabananas\$ p_1
a_4 nanas\$panama b_1
a_5 nas\$panamaba n_2
$\mathbf{a_6}$ s\$panamabana n_3
b_1 ananas\$panam a_1
m_1 abananas\$pan a_2
n_1 amabananas\$p a_3
n_2 anas\$panamab a_4
n_3 as\$panamaban a_5
p_1 anamabananas $\$_1$
s_1 \$panamabanan $\mathbf{a_6}$



Inverting BWT Again

$\$_1$ panamabanana s_1
a_1 bananas\$pana m_1
a_2 mabananas\$pa n_1
a_3 namabananas\$ p_1
a_4 nanas\$panama b_1
a_5 nas\$panamaba n_2
a_6 s\$panamabana n_3
b_1 ananas\$panam a_1
m_1 abananas\$pan a_2
n_1 amabananas\$p a_3
n_2 anas\$panamab a_4
n_3 as\$panamaban a_5
p_1 anamabananas $\$_1$
s_1 \$panamabanan a_6



Inverting BWT Again

$\$_1$panamabanana$s_1$

$a_1$bananas\$panam_1

$a_2$mabananas\$pan_1

$a_3$namabananas\$$p_1$

$a_4$nanas\$panamab_1

$a_5$nas\$panamaban_2

a_6s\$panamabana$\mathbf{n_3}$

$b_1$ananas\$panama_1

$m_1$abananas\$pana_2

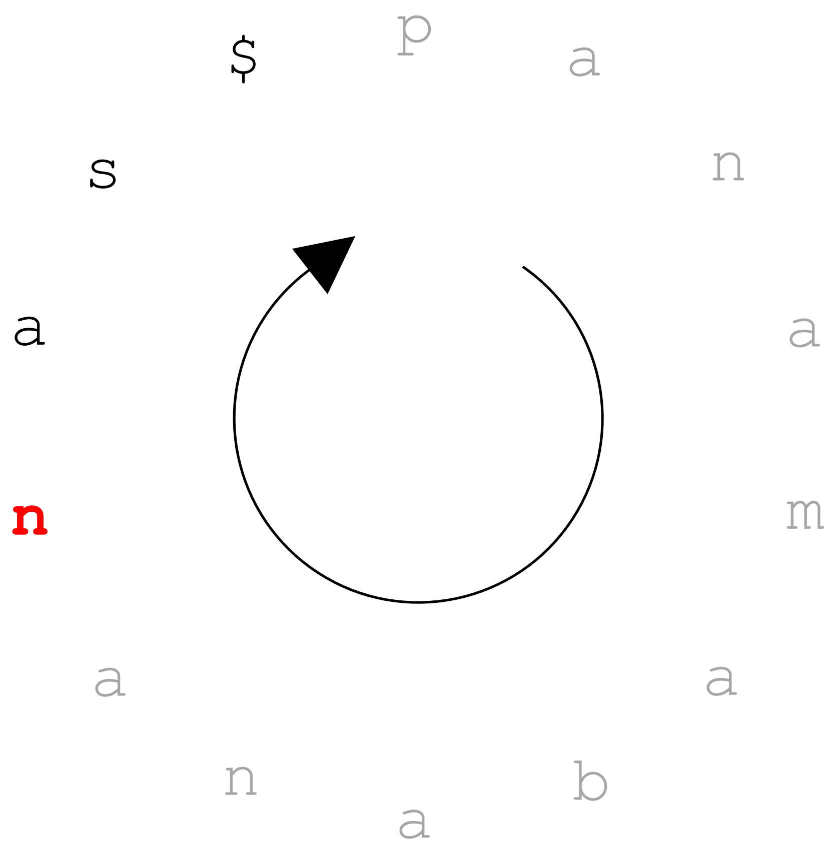
$n_1$amabananas\$pa_3

$n_2$anas\$panamaba_4

$\mathbf{n_3}$as\$panamabana_5

$p_1$anamabananas$\$_1$

s_1\$panamabanana_6



Inverting BWT Again

$\$_1$panamabanana$s_1$

$a_1$bananas\$panam_1

$a_2$mabananas\$pan_1

$a_3$namabananas\$$p_1$

$a_4$nanas\$panamab_1

$a_5$nas\$panamaban_2

a_6s\$panamabanan_3

$b_1$ananas\$panama_1

$m_1$abananas\$pana_2

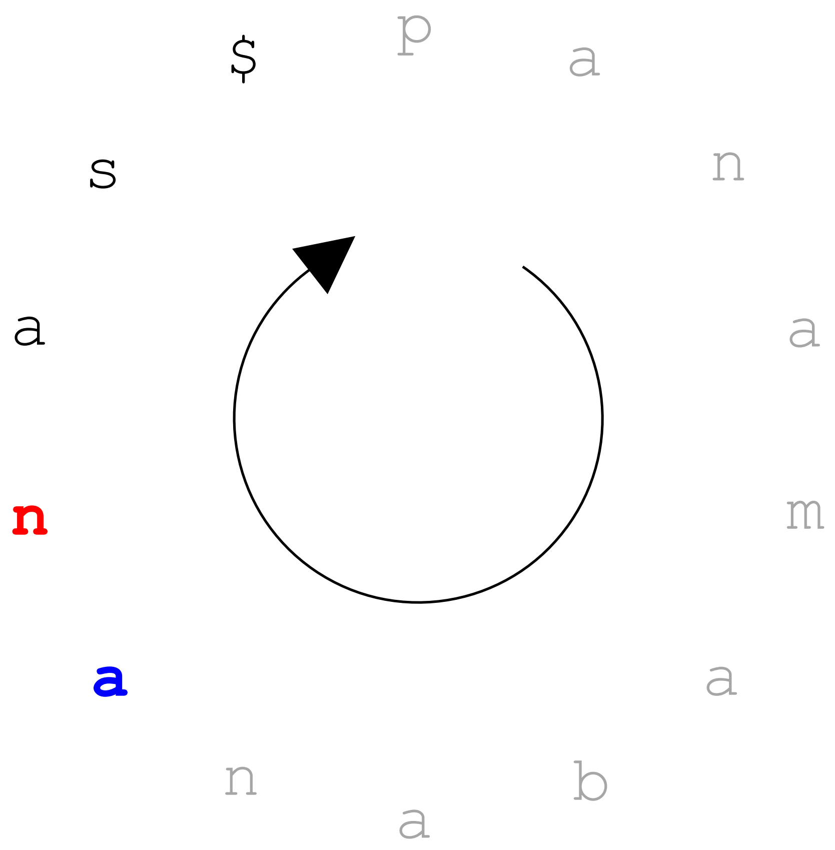
$n_1$amabananas\$pa_3

$n_2$anas\$panamaba_4

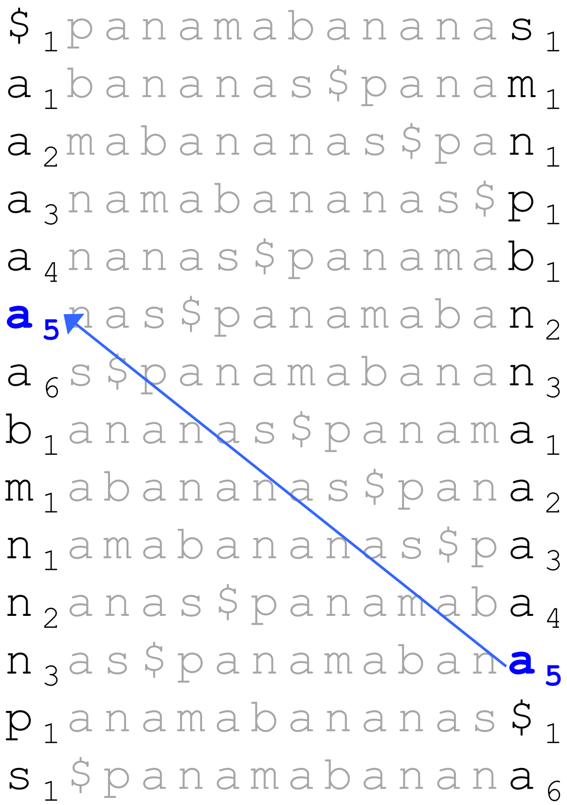
$\mathbf{n_3}$as\$panamaban$\mathbf{a_5}$

$p_1$anamabananas$\$_1$

s_1\$panamabanana_6



Inverting BWT Again



$\$_1$ panamabanana s_1

a_1 bananas\$pana m_1

a_2 mabananas\$pa n_1

a_3 namabananas\$ p_1

a_4 nanas\$panama b_1

a_5 nas\$panamaba n_2

a_6 s\$panamabana n_3

b_1 ananas\$panam a_1

m_1 abananas\$pan a_2

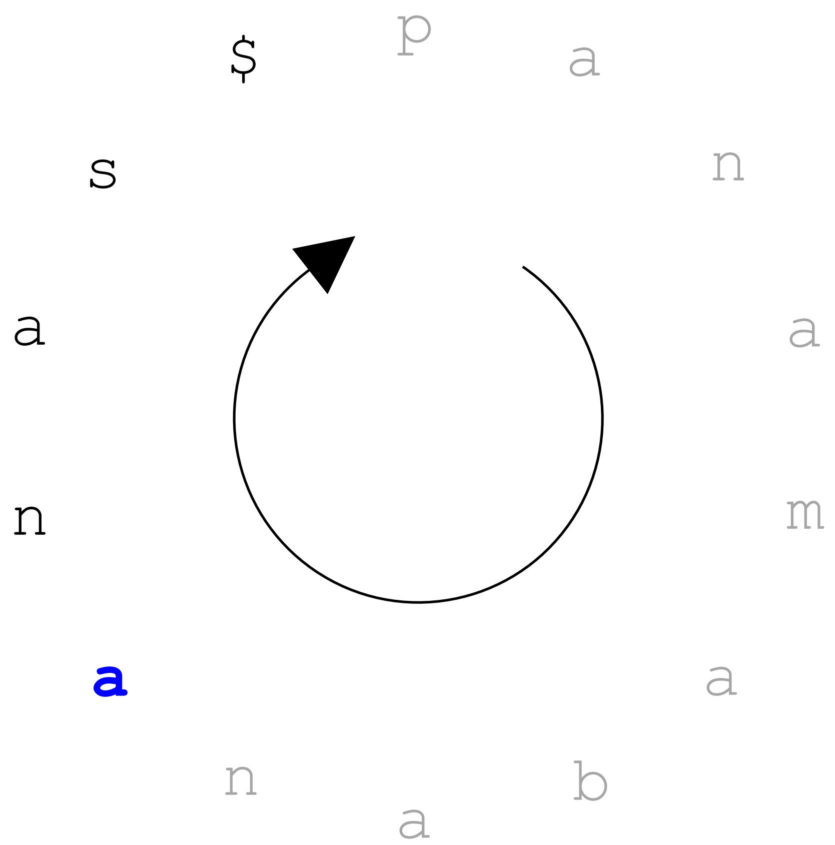
n_1 amabananas\$p a_3

n_2 anas\$panamab a_4

n_3 as\$panamaban a_5

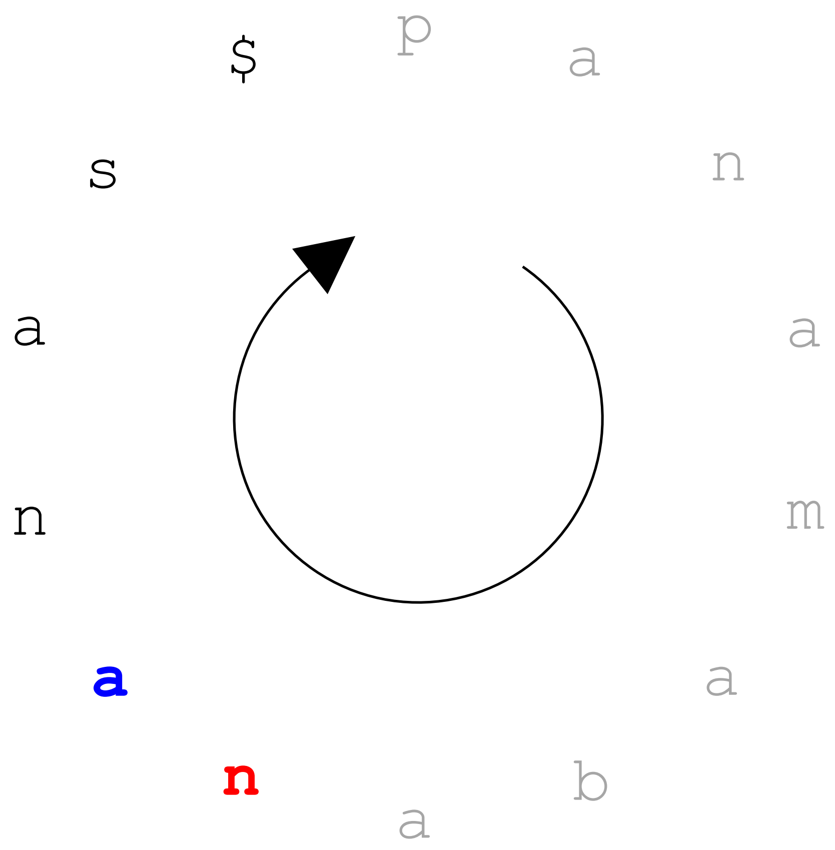
p_1 anamabananas $\$_1$

s_1 \$panamabanan a_6

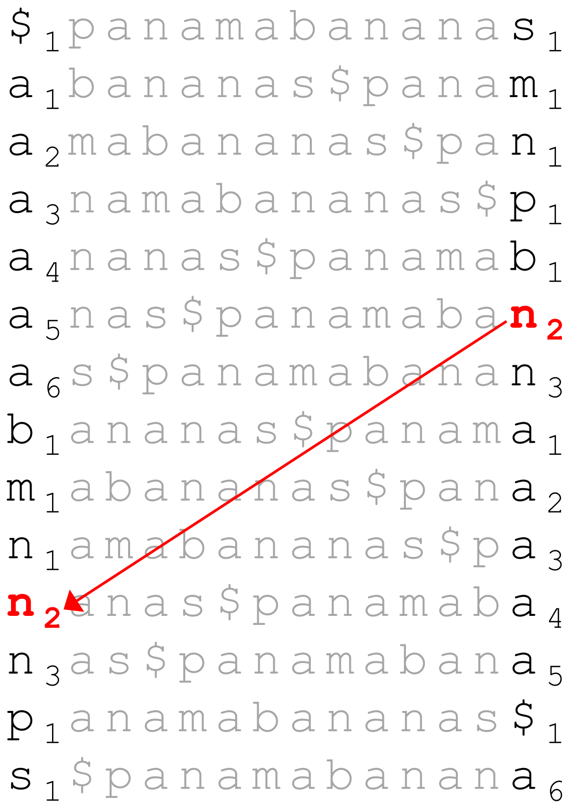


Inverting BWT Again

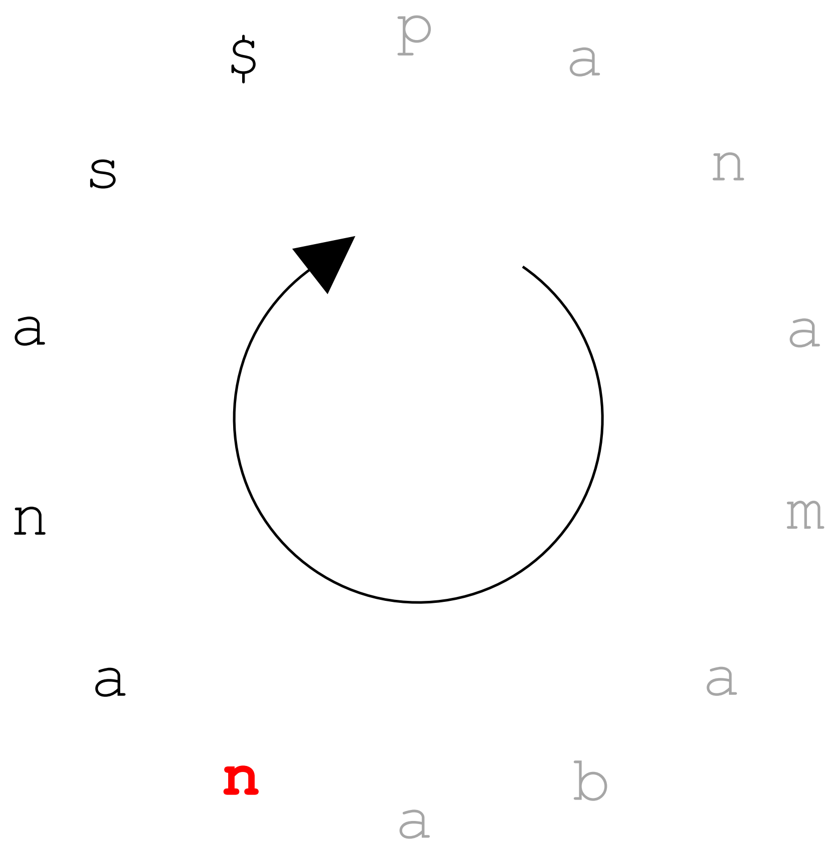
$\$_1$ panamabanana s_1
a_1 bananas\$pana m_1
a_2 mabananas\$pa n_1
a_3 namabananas\$ p_1
a_4 nanas\$panama b_1
$\mathbf{a_5}$ nas\$panamaba $\mathbf{n_2}$
a_6 s\$panamabana n_3
b_1 ananas\$panam a_1
m_1 abananas\$pan a_2
n_1 amabananas\$p a_3
n_2 anas\$panamab a_4
n_3 as\$panamaban a_5
p_1 anamabananas $\$_1$
s_1 \$panamabanan a_6



Inverting BWT Again



$\$_1$	panamabanana	s_1
a_1	bananas$pana	m_1
a_2	mabananas$pa	n_1
a_3	namabananas$	p_1
a_4	nanas$panama	b_1
a_5	nas$panamaba	**n_2**
a_6	s$panamabana	n_3
b_1	ananas$panam	a_1
m_1	abananas$pan	a_2
n_1	amabananas$p	a_3
n_2	anas$panamab	a_4
n_3	as$panamaban	a_5
p_1	anamabananas	$\$_1$
s_1	$panamabanan	a_6



Inverting BWT Again

$\$_1$panamabanana$s_1$

$a_1$bananas\$panam_1

$a_2$mabananas\$pan_1

$a_3$namabananas\$$p_1$

$a_4$nanas\$panamab_1

$a_5$nas\$panamaban_2

a_6s\$panamabanan_3

$b_1$ananas\$panama_1

$m_1$abananas\$pana_2

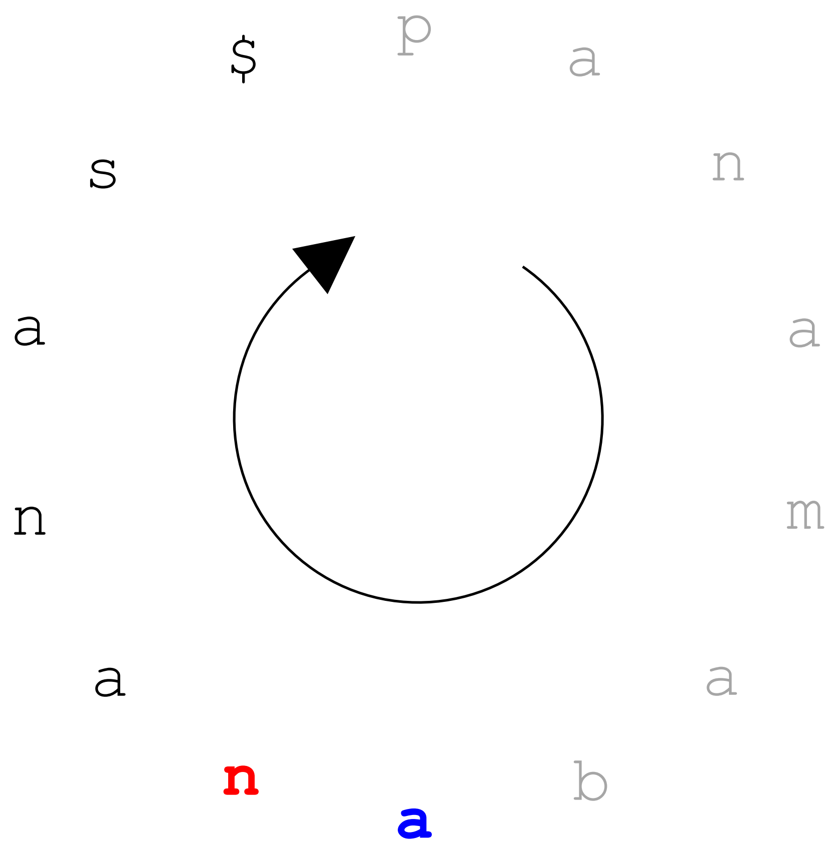
$n_1$amabananas\$pa_3

$n_2$anas\$panamab**a_4**

$n_3$as\$panamabana_5

$p_1$anamabananas$\$_1$

s_1\$panamabanana_6



$\$_1$ panamabanana s_1

a_1 bananas\$pana m_1

a_2 mabananas\$pa n_1

a_3 namabananas\$ p_1

a_4 nanas\$panama b_1

a_5 nas\$panamaba n_2

a_6 s\$panamabana n_3

b_1 ananas\$panam a_1

m_1 abananas\$pan a_2

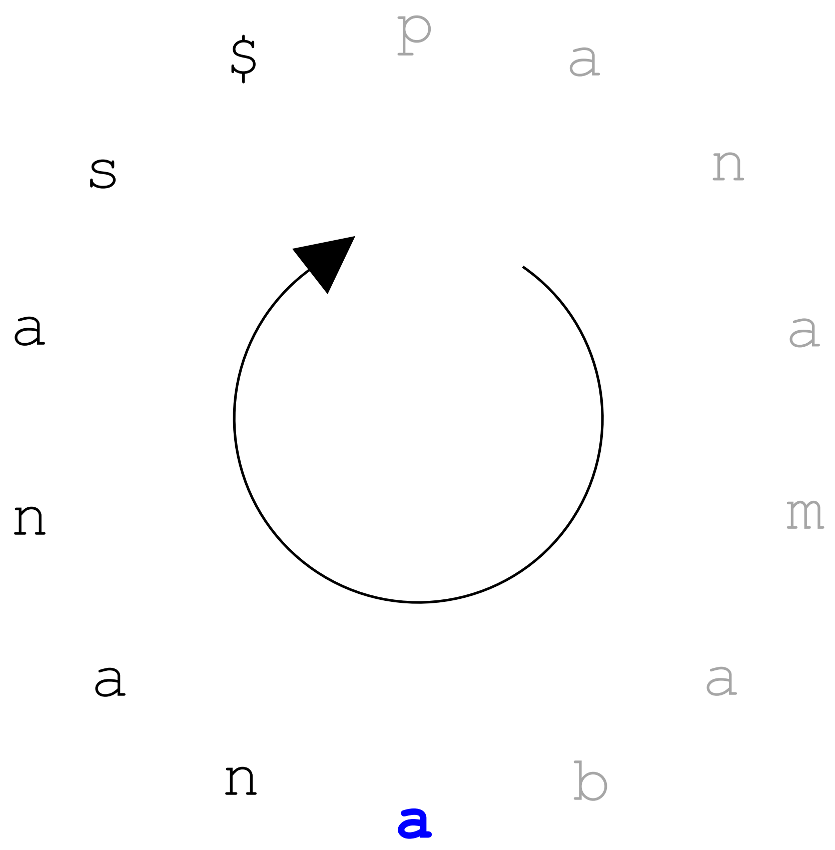
n_1 amabananas\$p a_3

n_2 anas\$panamab **a_4**

n_3 as\$panamaban a_5

p_1 anamabananas $\$_1$

s_1 \$panamabanan a_6



$\$_1\text{panamabanana}s_1$

$a_1\text{bananas\$pana}m_1$

$a_2\text{mabananas\$pa}n_1$

$a_3\text{namabananas\$}p_1$

$a_4\text{nanas\$panama}b_1$

$a_5\text{nas\$panamaba}n_2$

$a_6\text{s\$panamabana}n_3$

$b_1\text{ananas\$panam}a_1$

$m_1\text{abananas\$pan}a_2$

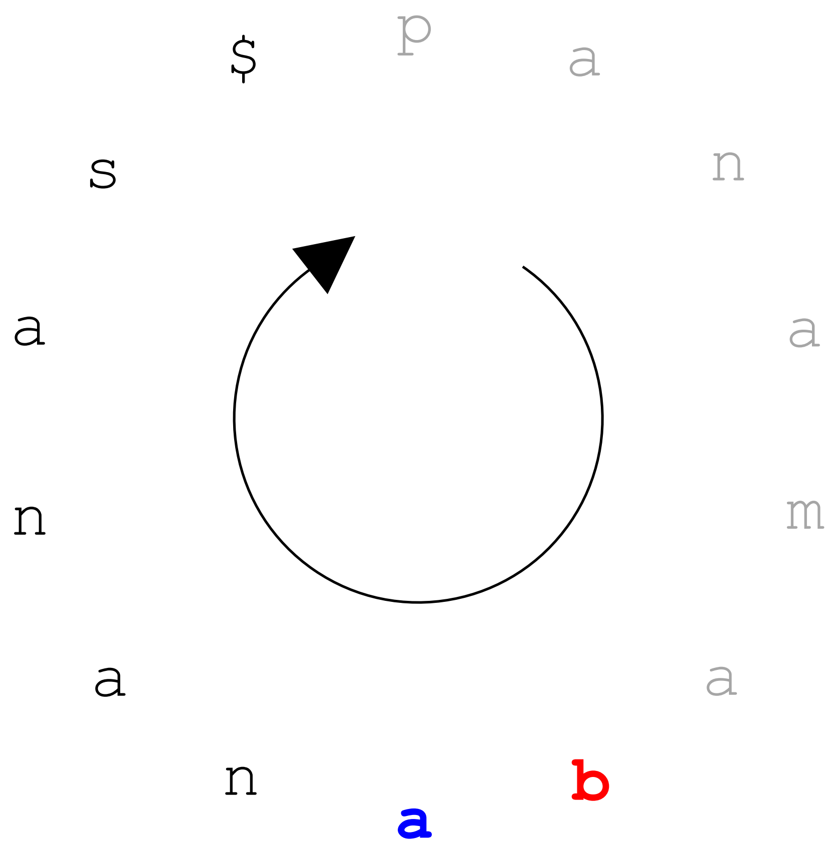
$n_1\text{amabananas\$p}a_3$

$n_2\text{anas\$panamab}a_4$

$n_3\text{as\$panamaban}a_5$

$p_1\text{anamabananas}\$_1$

$s_1\text{\$panamabanan}a_6$



Inverting BWT Again

$\$_1$ panamabanana s_1

a_1 bananas\$pana m_1

a_2 mabananas\$pa n_1

a_3 namabananas\$ p_1

a_4 nanas\$panama b_1

a_5 nas\$panamaba n_2

a_6 s\$panamabana n_3

b_1 ananas\$panam a_1

m_1 abananas\$pan a_2

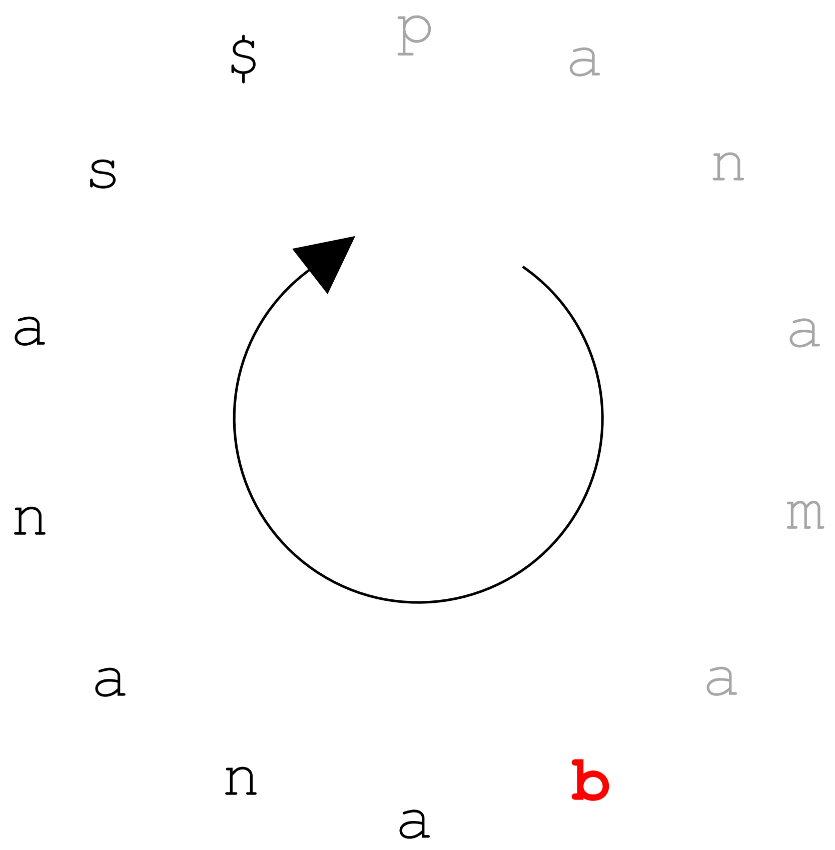
n_1 amabananas\$p a_3

n_2 anas\$panamab a_4

n_3 as\$panamaban a_5

p_1 anamabananas $\$_1$

s_1 \$panamabanan a_6



Inverting BWT Again

$\$_1$panamabanana$s_1$

$a_1$bananas\$panam_1

$a_2$mabananas\$pan_1

$a_3$namabananas\$$p_1$

$a_4$nanas\$panamab_1

$a_5$nas\$panamaban_2

a_6s\$panamabanan_3

$b_1$ananas\$panama_1

$m_1$abananas\$pana_2

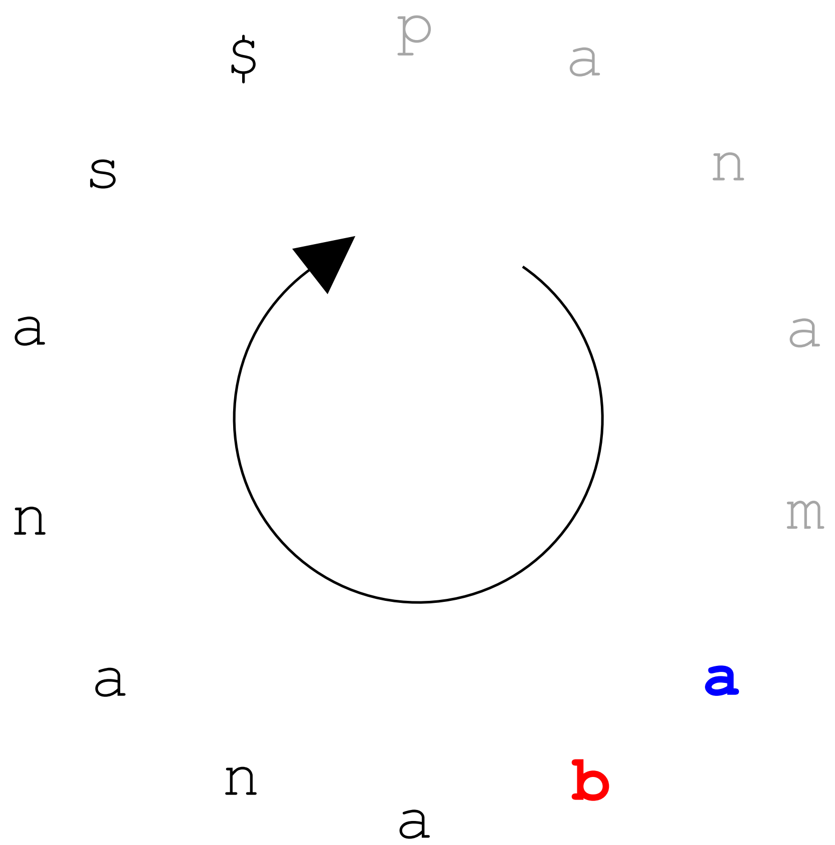
$n_1$amabananas\$pa_3

$n_2$anas\$panamaba_4

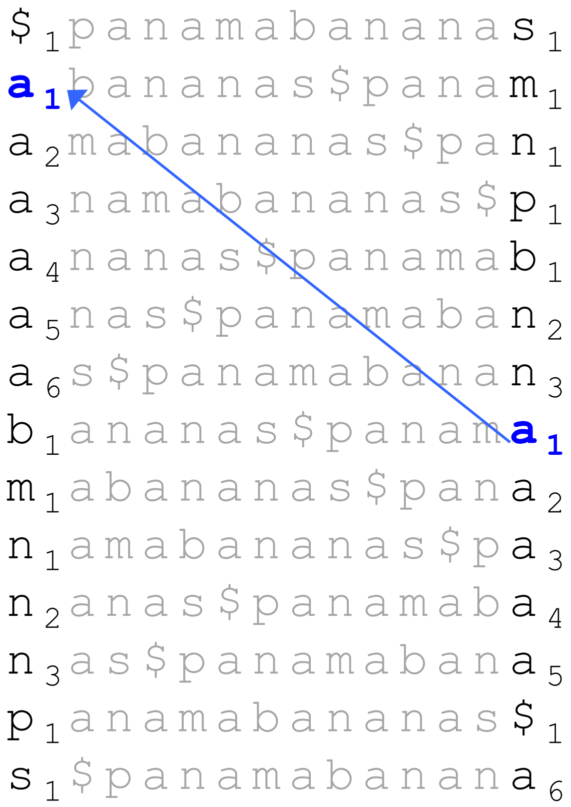
$n_3$as\$panamabana_5

$p_1$anamabananas$\$_1$

s_1\$panamabanana_6



Inverting BWT Again



$\$_1$ panamabanana s_1

a_1 bananas\$pana m_1

a_2 mabananas\$pa n_1

a_3 namabananas\$ p_1

a_4 nanas\$panama b_1

a_5 nas\$panamaba n_2

a_6 s\$panamabana n_3

b_1 ananas\$panam a_1

m_1 abananas\$pan a_2

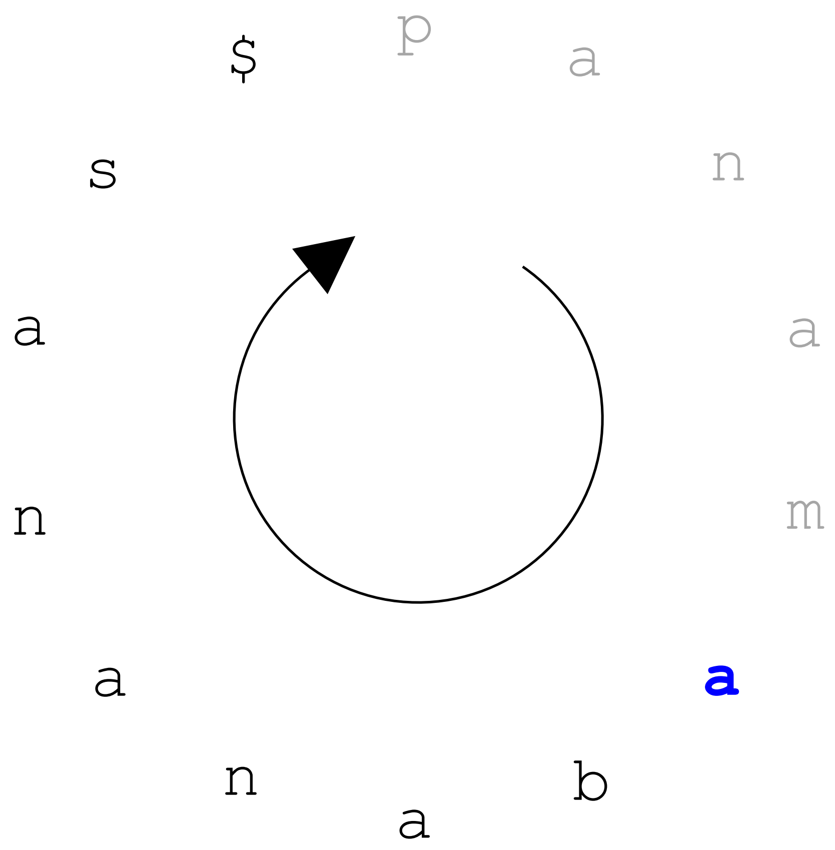
n_1 amabananas\$p a_3

n_2 anas\$panamab a_4

n_3 as\$panamaban a_5

p_1 anamabananas $\$_1$

s_1 \$panamabanan a_6



Inverting BWT Again

$\$_1$ panamabanana s_1

a_1 bananas$pana m_1

a_2 mabananas$pa n_1

a_3 namabananas$ p_1

a_4 nanas$panama b_1

a_5 nas$panamaba n_2

a_6 s$panamabana n_3

b_1 ananas$panam a_1

m_1 abananas$pan a_2

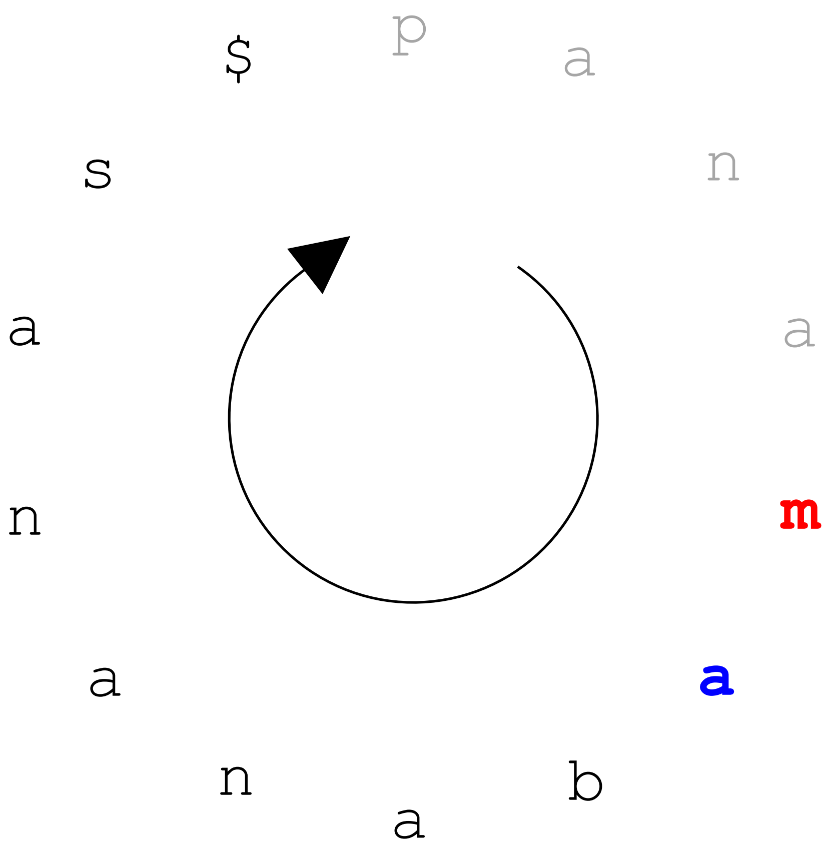
n_1 amabananas$p a_3

n_2 anas$panamab a_4

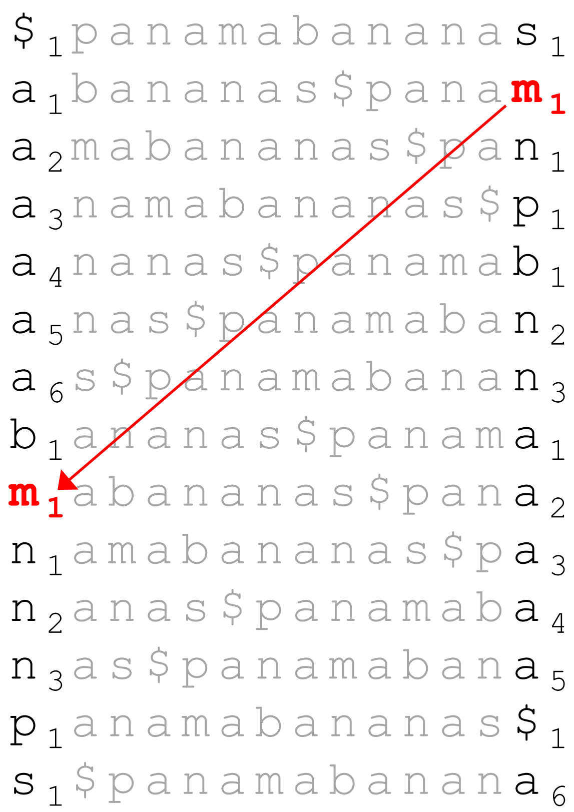
n_3 as$panamaban a_5

p_1 anamabananas $\$_1$

s_1 $panamabanan a_6



Inverting BWT Again



$\$_{1}\,\text{panamabanana}\,s_{1}$

$a_{1}\,\text{bananas\$pana}\,m_{1}$

$a_{2}\,\text{mabananas\$pa}\,n_{1}$

$a_{3}\,\text{namabananas\$}\,p_{1}$

$a_{4}\,\text{nanas\$panama}\,b_{1}$

$a_{5}\,\text{nas\$panamaba}\,n_{2}$

$a_{6}\,\text{s\$panamabana}\,n_{3}$

$b_{1}\,\text{ananas\$panam}\,a_{1}$

$m_{1}\,\text{abananas\$pan}\,a_{2}$

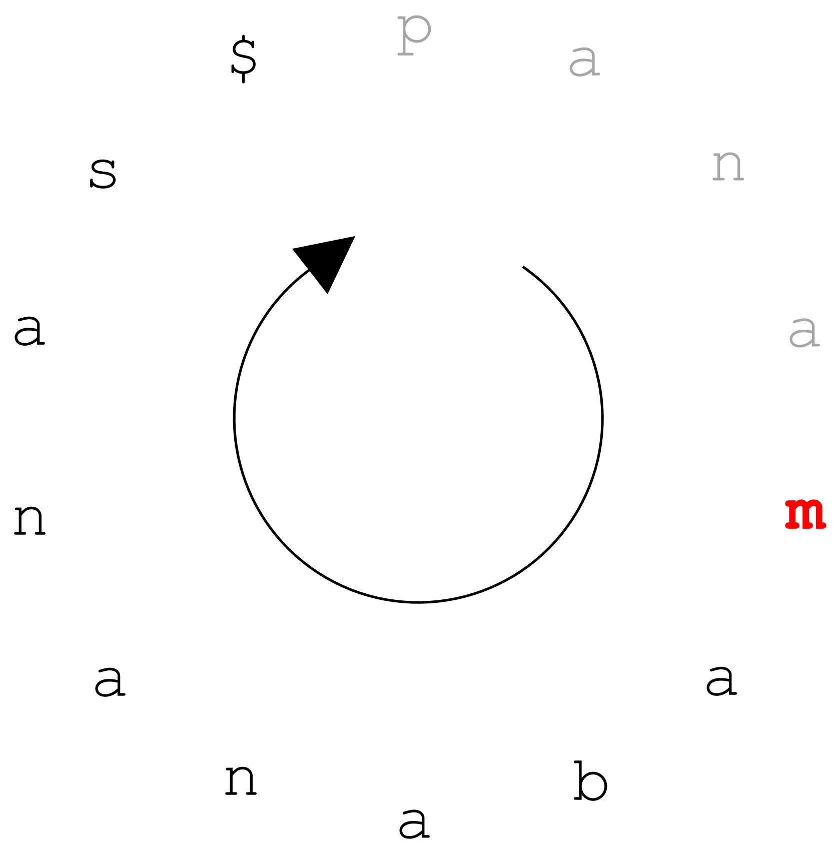
$n_{1}\,\text{amabananas\$p}\,a_{3}$

$n_{2}\,\text{anas\$panamab}\,a_{4}$

$n_{3}\,\text{as\$panamaban}\,a_{5}$

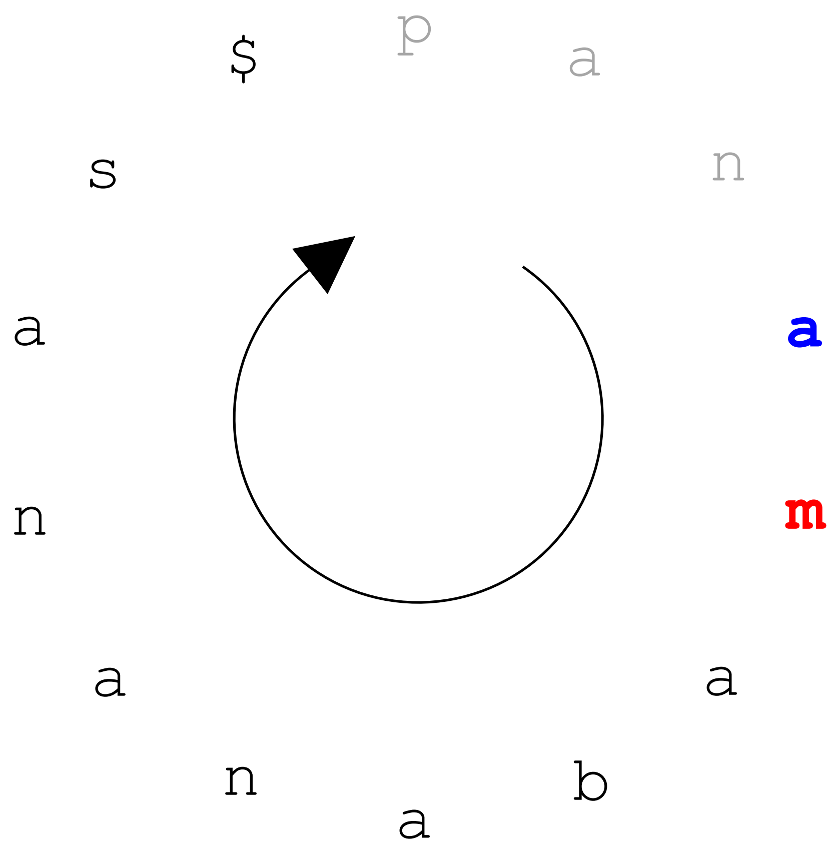
$p_{1}\,\text{anamabananas}\,\$_{1}$

$s_{1}\,\text{\$panamabanan}\,a_{6}$

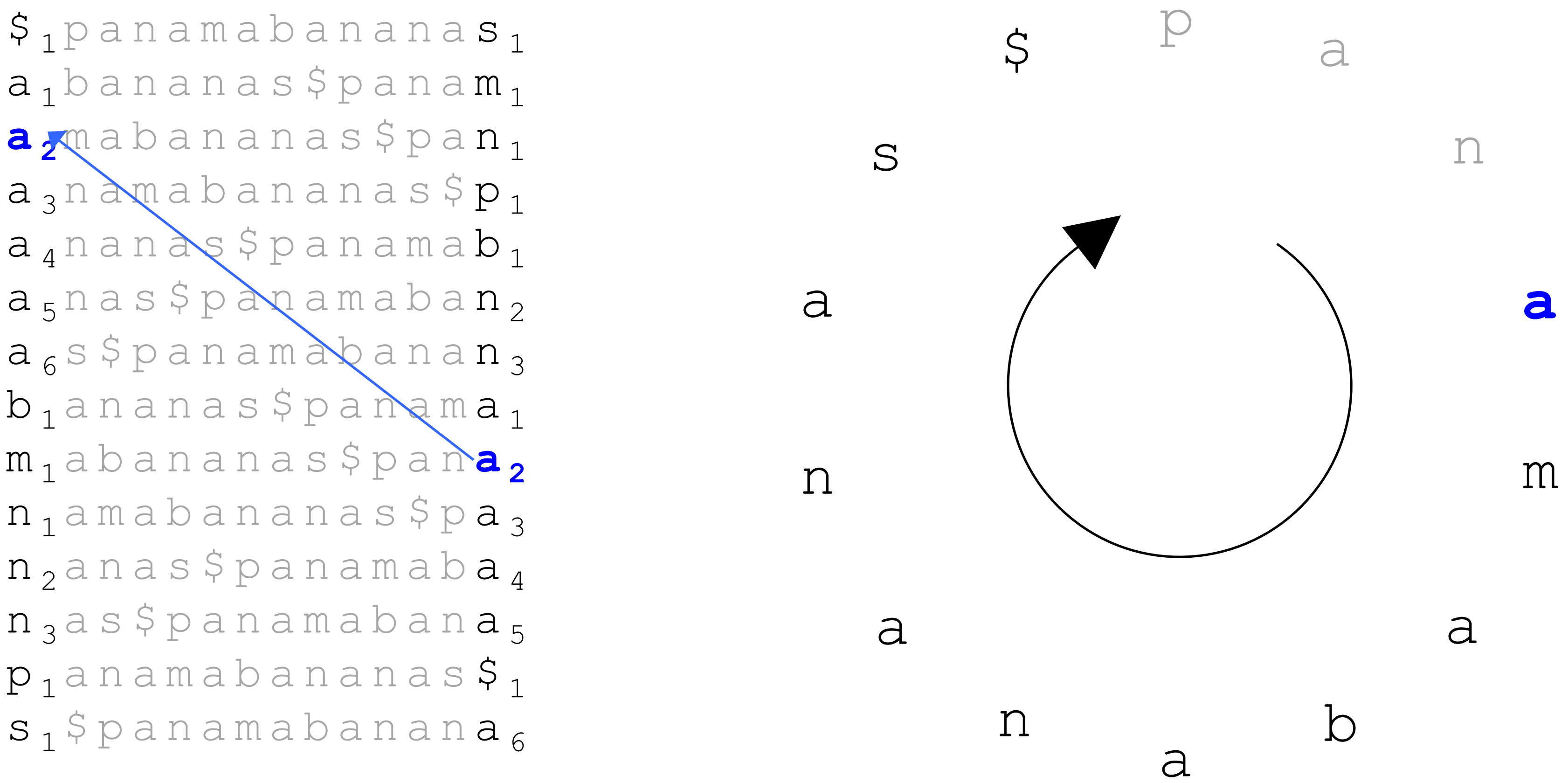


Inverting BWT Again

$\$_1$ panamabanana s_1
a_1 bananas$pana m_1
a_2 mabananas$pa n_1
a_3 namabananas$ p_1
a_4 nanas$panama b_1
a_5 nas$panamaba n_2
a_6 s$panamabana n_3
b_1 ananas$panam a_1
m_1 abananas$pan **$a_2$**
n_1 amabananas$p a_3
n_2 anas$panamab a_4
n_3 as$panamaban a_5
p_1 anamabananas $\$_1$
s_1 $panamabanan a_6



Inverting BWT Again



$\$_1$ panamabanana s_1

a_1 bananas\$pana m_1

a_2 mabananas\$pa n_1

a_3 namabananas\$ p_1

a_4 nanas\$panama b_1

a_5 nas\$panamaba n_2

a_6 s\$panamabana n_3

b_1 ananas\$panam a_1

m_1 abananas\$pan a_2

n_1 amabananas\$p a_3

n_2 anas\$panamab a_4

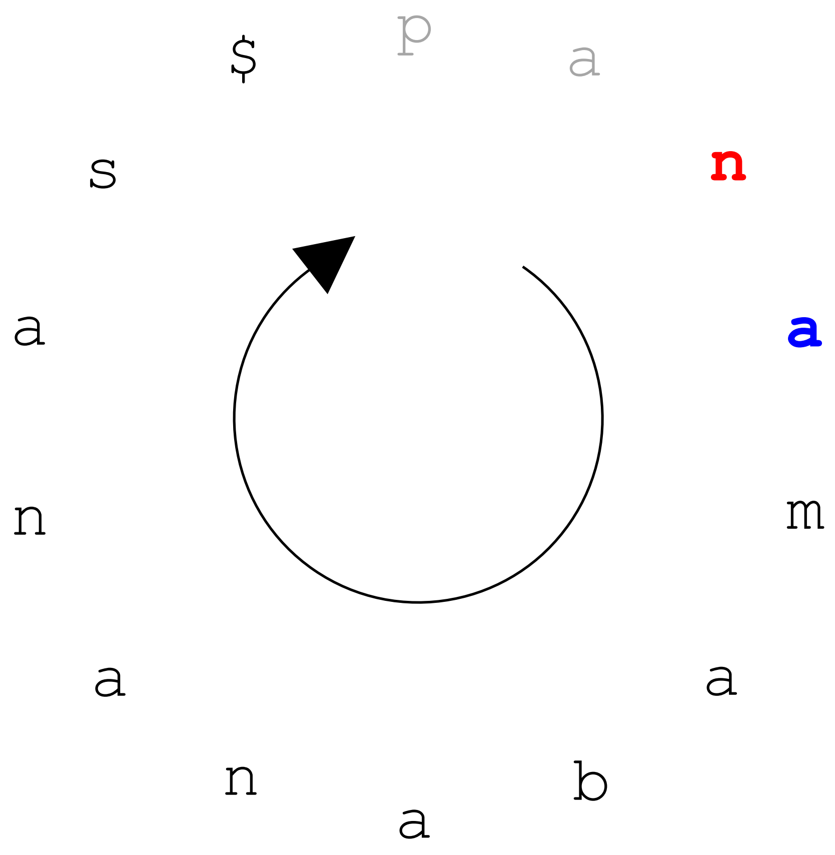
n_3 as\$panamaban a_5

p_1 anamabananas $\$_1$

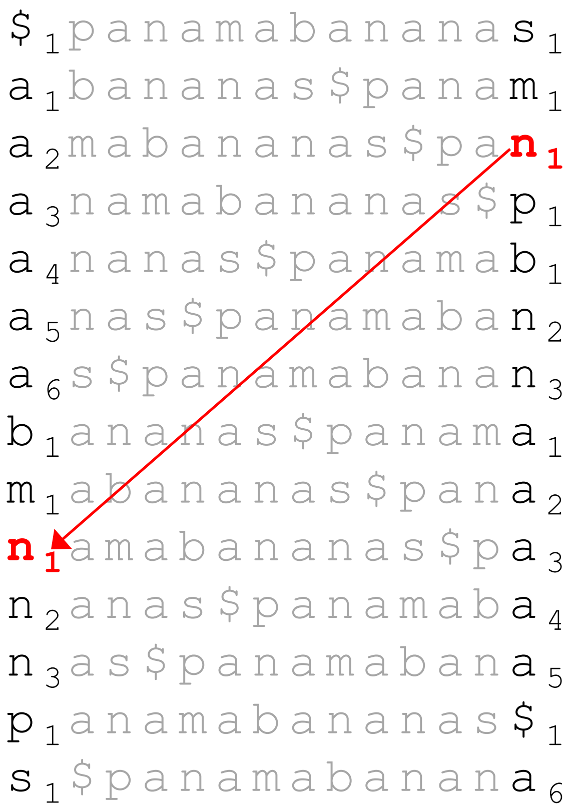
s_1 \$panamabanan a_6

Inverting BWT Again

$\$_1$panamabanana$s_1$
$a_1$bananas\$panamm_1
$a_2$mabananas\$pa**n_1**
$a_3$namabananas\$$p_1$
$a_4$nanas\$panamab_1
$a_5$nas\$panamaban_2
a_6s\$panamabanan_3
$b_1$ananas\$panama_1
$m_1$abananas\$pana_2
$n_1$amabananas\$pa_3
$n_2$anas\$panamaba_4
$n_3$as\$panamabana_5
$p_1$anamabananas$\$_1$
s_1\$panamabanana_6



Inverting BWT Again



$\$_1$ panamabanana s_1

a_1 bananas\$pana m_1

a_2 mabananas\$pa n_1

a_3 namabananas\$ p_1

a_4 nanas\$panama b_1

a_5 nas\$panamaba n_2

a_6 s\$panamabana n_3

b_1 ananas\$panam a_1

m_1 abananas\$pan a_2

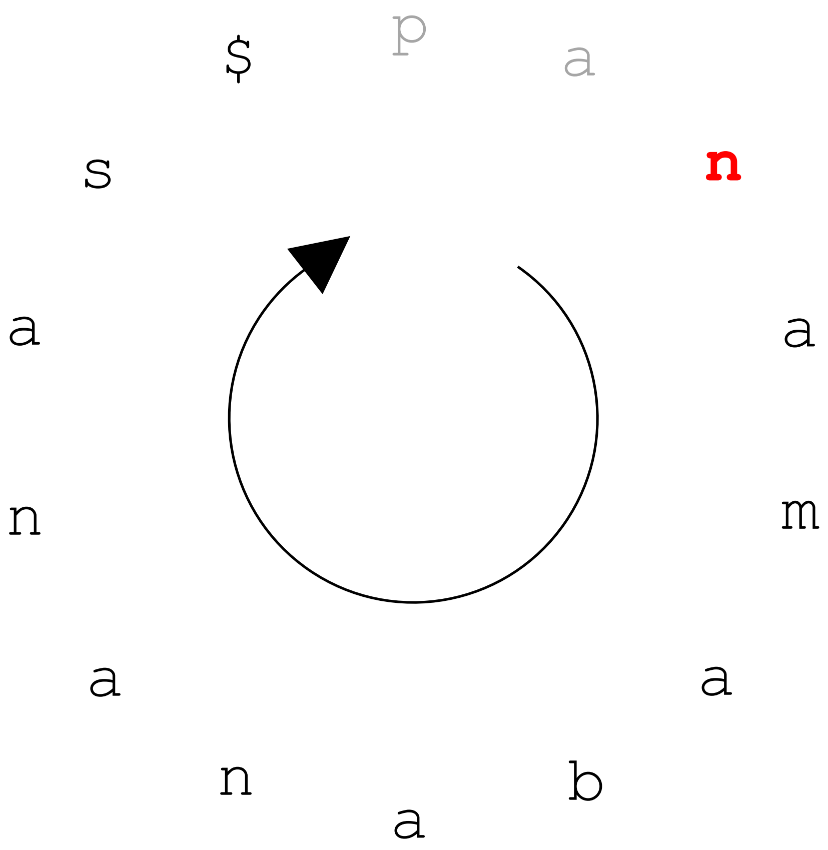
n_1 amabananas\$p a_3

n_2 anas\$panamab a_4

n_3 as\$panamaban a_5

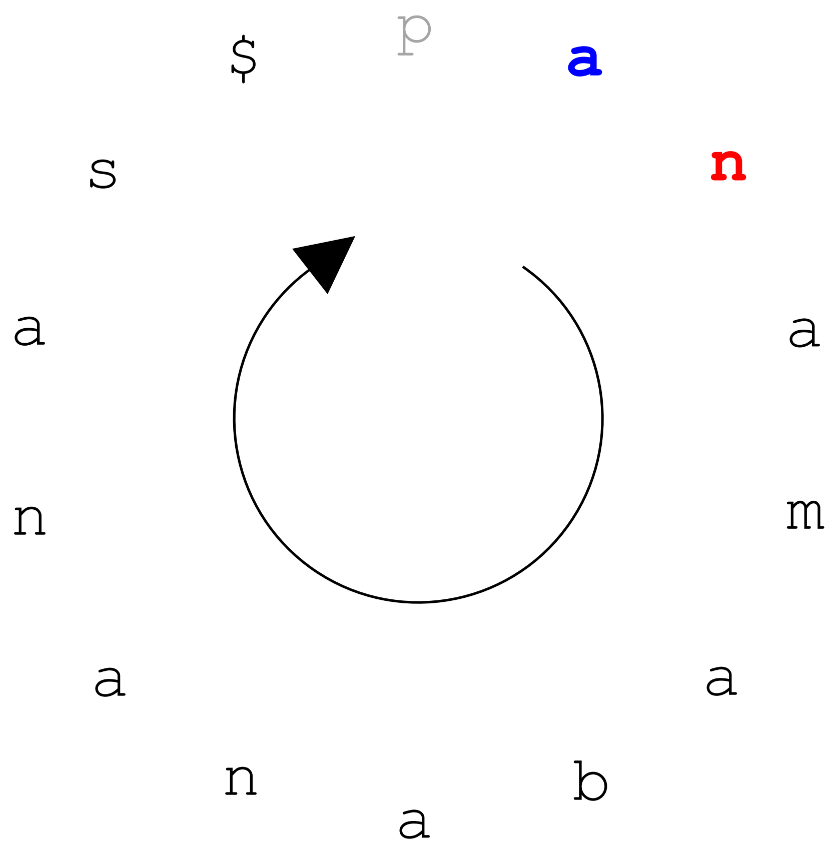
p_1 anamabananas $\$_1$

s_1 \$panamabanan a_6

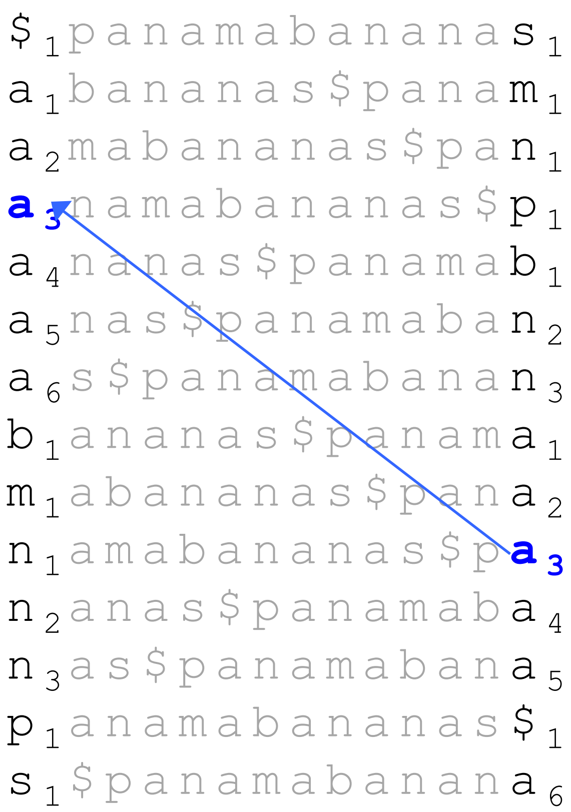


Inverting BWT Again

$\$_1$panamabanana$s_1$
$a_1$bananas\$panam_1
$a_2$mabananas\$pan_1
$a_3$namabananas\$$p_1$
$a_4$nanas\$panamab_1
$a_5$nas\$panamaban_2
a_6s\$panamabanan_3
$b_1$ananas\$panama_1
$m_1$abananas\$pana_2
$\mathbf{n_1}$amabananas\$p$\mathbf{a_3}$
$n_2$anas\$panamaba_4
$n_3$as\$panamabana_5
$p_1$anamabananas$\$_1$
s_1\$panamabanana_6



Inverting BWT Again



$\$_1$ panamabanana s_1

a_1 bananas\$pana m_1

a_2 mabananas\$pa n_1

a_3 namabananas\$ p_1

a_4 nanas\$panama b_1

a_5 nas\$panamaba n_2

a_6 s\$panamabana n_3

b_1 ananas\$panam a_1

m_1 abananas\$pan a_2

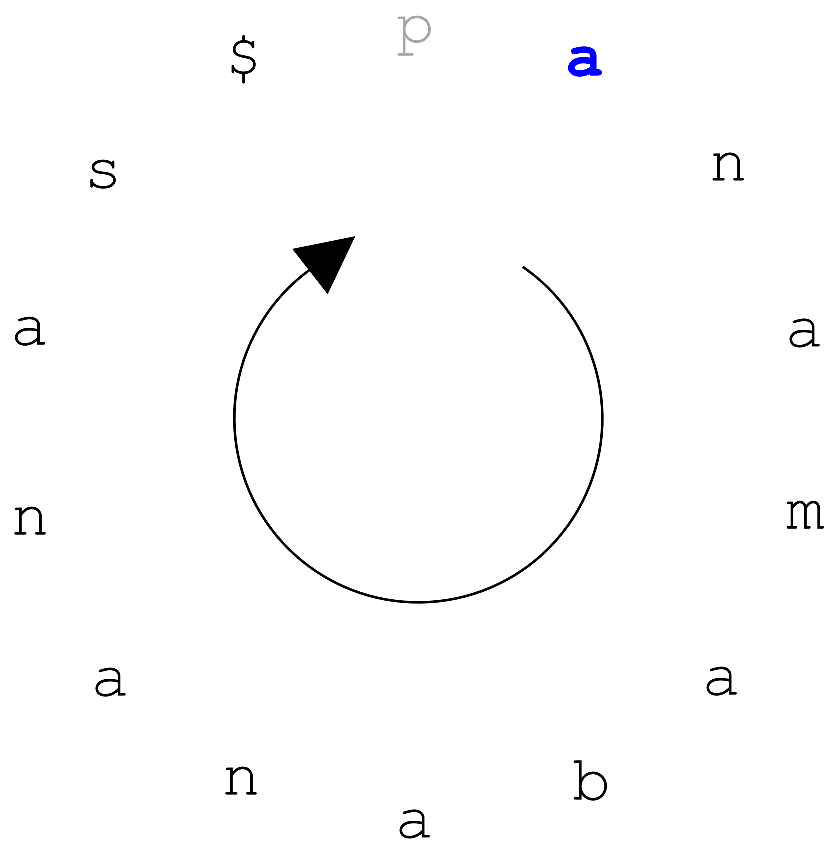
n_1 amabananas\$p a_3

n_2 anas\$panamab a_4

n_3 as\$panamaban a_5

p_1 anamabananas $\$_1$

s_1 \$panamabanan a_6



Inverting BWT Again

$\$_{1}$panamabanana$s_{1}$

a_{1}bananas\$panam_{1}

a_{2}mabananas\$pan_{1}

a_{3}namabananas\$**p_{1}**

a_{4}nanas\$panamab_{1}

a_{5}nas\$panamaban_{2}

a_{6}s\$panamabanan_{3}

b_{1}ananas\$panama_{1}

m_{1}abananas\$pana_{2}

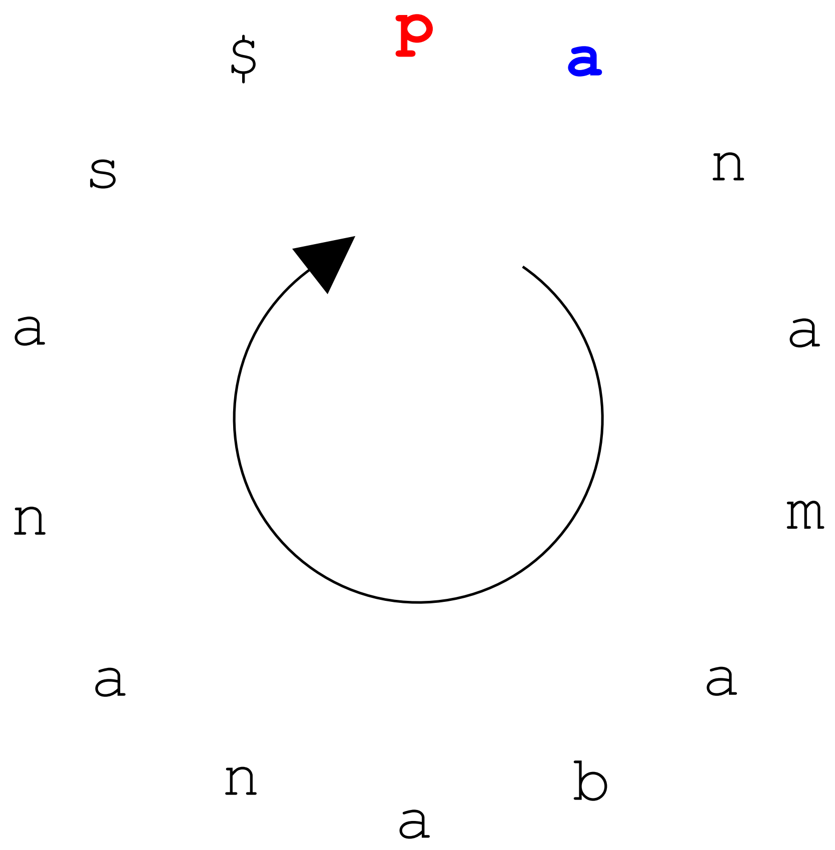
n_{1}amabananas\$pa_{3}

n_{2}anas\$panamaba_{4}

n_{3}as\$panamabana_{5}

p_{1}anamabananas$\$_{1}$

s_{1}\$panamabanana_{6}



Inverting BWT Again

$\$_1$ panamabanana s_1

a_1 bananas\$pana m_1

a_2 mabananas\$pa n_1

a_3 namabananas\$ p_1

a_4 nanas\$panama b_1

a_5 nas\$panamaba n_2

a_6 s\$panamabana n_3

b_1 ananas\$panam a_1

m_1 abananas\$pan a_2

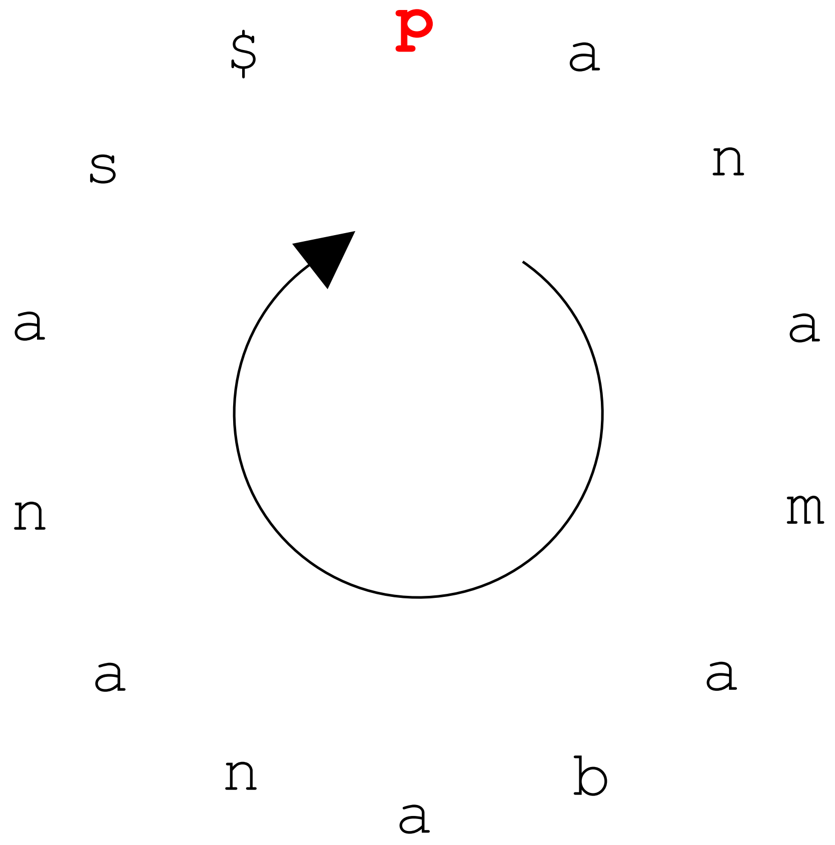
n_1 amabananas\$p a_3

n_2 anas\$panamab a_4

n_3 as\$panamaban a_5

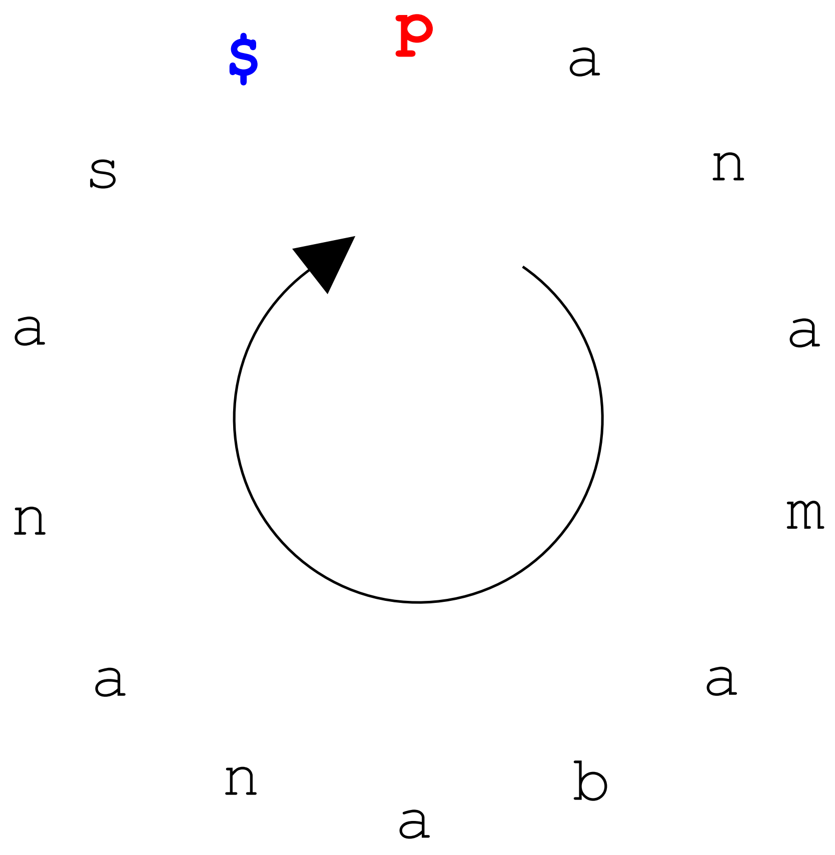
p_1 anamabananas $\$_1$

s_1 \$panamabanan a_6

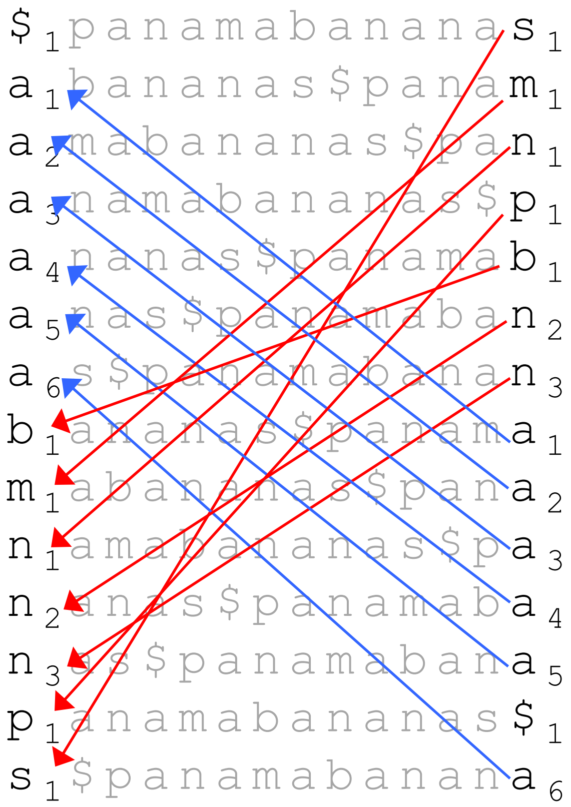


We Are Done!

$\$_1$ panamabanana s_1
a_1 bananas\$pana m_1
a_2 mabananas\$pa n_1
a_3 namabananas\$ p_1
a_4 nanas\$panama b_1
a_5 nas\$panamaba n_2
a_6 s\$panamabana n_3
b_1 ananas\$panam a_1
m_1 abananas\$pan a_2
n_1 amabananas\$p a_3
n_2 anas\$panamab a_4
n_3 as\$panamaban a_5
p_1 anamabananas $\$_1$
s_1 \$panamabanan a_6



This Was Fast!



$\$_1$ panamabanana s_1

a_1 bananas\$pana m_1

a_2 mabananas\$pan n_1

a_3 namabananas\$ p_1

a_4 nanas\$panama b_1

a_5 nas\$panamaba n_2

a_6 s\$panamabana n_3

b_1 ananas\$panam a_1

m_1 abananas\$pan a_2

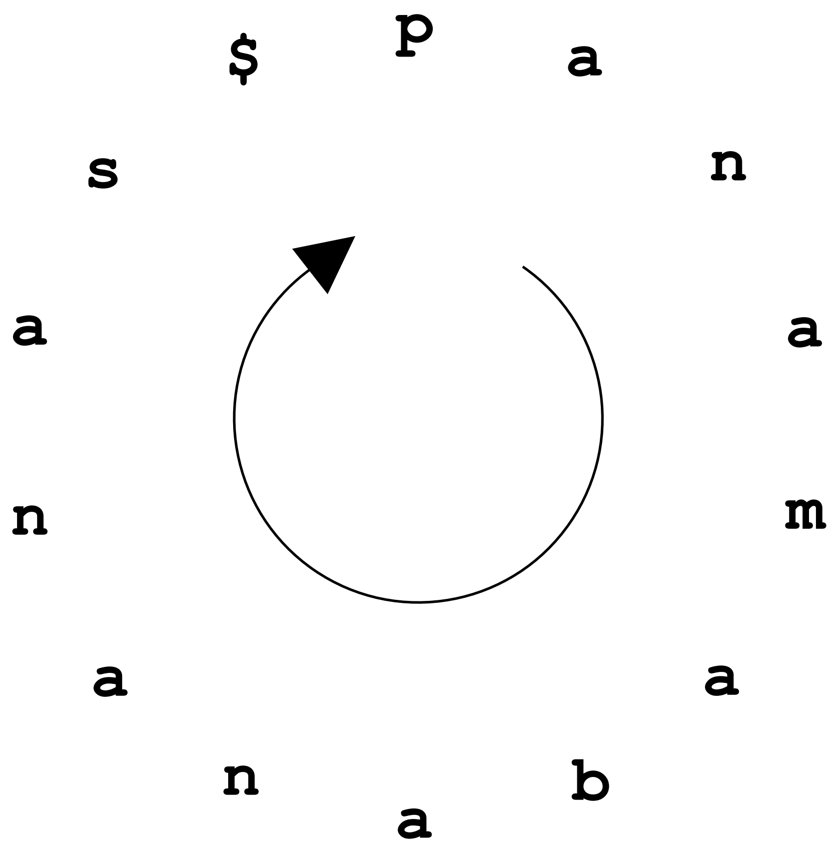
n_1 amabananas\$p a_3

n_2 anas\$panamab a_4

n_3 as\$panamaban a_5

p_1 anamabananas $\$_1$

s_1 \$panamabanan a_6



- Memory: 2|*Text*|
- Time: O(|*Text*|)

Outline

- Burrows-Wheeler Transform
- Inverting Burrows-Wheeler Transform
- **Using BWT for Pattern Matching**
- Suffix Arrays
- Approximate Pattern Matching

Back to Pattern Matching

- Suffix Tree Pattern Matching:
  - Runtime: O(|*Text*| + |*Patterns*|)
  - Memory: 20•|*Text*|

For human genome:

- |*Text*| ≈ $3*10^9$

- Can we use BWT(*Text*) to design a more memory efficient linear-time algorithm for Multiple Pattern Matching?

Notice that initially we were using burrow wheels algorithm to just create runs from repeats but now we are posing a different question itself that is can we design a more memory efficient linear time algorithm for multiple pattern matching using BWT as we know that our best method Suffix Tree Pattern Matching is taking 20|Text| of memory

Finding Pattern Matches Using BWT

- Searching for **ana** in p**ana**mab**anana**s

$\$_1$panamabanana$s_1$

$a_1$bananas\$panam_1

$a_2$mabananas\$pan_1

$a_3$namabananas\$$p_1$

$a_4$nanas\$panamab_1

$a_5$nas\$panamaban_2

a_6s\$panamabanan_3

$b_1$ananas\$panama_1

$m_1$abananas\$pana_2

$n_1$amabananas\$pa_3

$n_2$anas\$panamaba_4

$n_3$as\$panamabana_5

$p_1$anamabananas$\$_1$

s_1\$panamabanana_6

Lets Start by Matching the Last Symbol (a)

- Searching for an**a** in panamabananas

$\$_1$	panamabanana	s_1
a_1	bananas$pana	m_1
a_2	mabananas$pa	n_1
a_3	namabananas$	p_1
a_4	nanas$panama	b_1
a_5	nas$panamaba	n_2
a_6	s$panamabana	n_3
b_1	ananas$panam	a_1
m_1	abananas$pan	a_2
n_1	amabananas$p	a_3
n_2	anas$panamab	a_4
n_3	as$panamaban	a_5
p_1	anamabananas	$\$_1$
s_1	$panamabanan	a_6

Matching the Last Two Symbols (na)

- Searching for ana in panamabananas

$\$_1\text{panamabanana}s_1$

$a_1\text{bananas\$pana}m_1$

$a_2\text{mabananas\$pa}n_1$

$a_3\text{namabananas\$}p_1$

$a_4\text{nanas\$panama}b_1$

$a_5\text{nas\$panamaba}n_2$

$a_6\text{s\$panamabana}n_3$

$b_1\text{ananas\$panam}a_1$

$m_1\text{abananas\$pan}a_2$

$n_1\text{amabananas\$p}a_3$

$n_2\text{anas\$panamab}a_4$

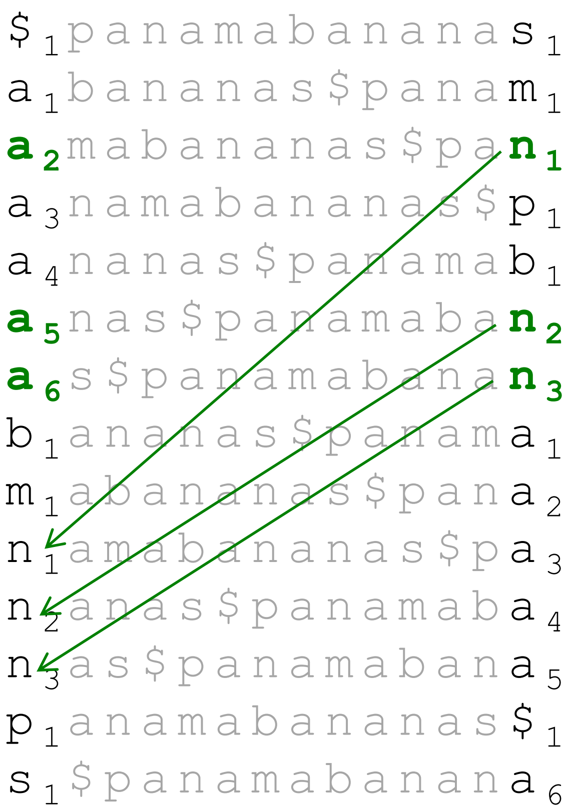
$n_3\text{as\$panamaban}a_5$

$p_1\text{anamabananas}\$_1$

$s_1\text{\$panamabanan}a_6$

Three Matches of na Found!

- Searching for a**na** in panamabananas



$\$_1$panamabanana$s_1$

$a_1$bananas\$panam_1

$\mathbf{a_2}$mabananas\$pa$\mathbf{n_1}$

$a_3$namabananas\$$p_1$

$a_4$nanas\$panamab_1

$\mathbf{a_5}$nas\$panamaba$\mathbf{n_2}$

$\mathbf{a_6}$s\$panamabana$\mathbf{n_3}$

$b_1$ananas\$panama_1

$m_1$abananas\$pana_2

$n_1$amabananas\$pa_3

$n_2$anas\$panamaba_4

$n_3$as\$panamabana_5

$p_1$anamabananas$\$_1$

s_1\$panamabanana_6

Three Matches of **na** Found!

- Searching for a**na** in panamabananas

```
$1 panamabanana s1
a1 bananas$pana m1
a2 mabananas$pa n1
a3 namabananas$ p1
a4 nanas$panama b1
a5 nas$panamaba n2
a6 s$panamabana n3
b1 ananas$panam a1
m1 abananas$pan a2
n1 amabananas$p a3
n2 anas$panamab a4
n3 as$panamaban a5
p1 anamabananas $1
s1 $panamabanan a6
```

Three Matches of **na** Found!

- Searching for a**na** in panamabananas

$\$_1$ panamabanana s_1

a_1 bananas$pana m_1

a_2 mabananas$pa n_1

a_3 namabananas$ p_1

a_4 nanas$panama b_1

a_5 nas$panamaba n_2

a_6 s$panamabana n_3

b_1 ananas$panam a_1

m_1 abananas$pan a_2

n_1a mabananas$p a_3

n_2a nas$panamab a_4

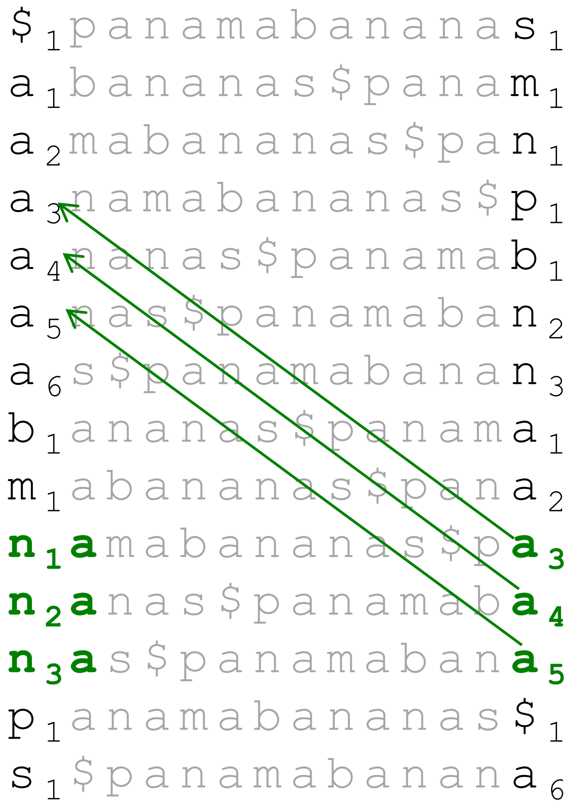
n_3a s$panamaban a_5

p_1 anamabananas $\$_1$

s_1 $panamabanan a_6

Matching **ana**

- Searching for **ana** in panamabananas



$\$_1$ panamabanana s_1

a_1 bananas\$pana m_1

a_2 mabananas\$pa n_1

a_3 namabananas\$ p_1

a_4 nanas\$panama b_1

a_5 nas\$panamaba n_2

a_6 s\$panamabana n_3

b_1 ananas\$panam a_1

m_1 abananas\$pan a_2

n_1 amabananas\$p **a_3**

n_2 anas\$panamab **a_4**

n_3 as\$panamaban **a_5**

p_1 anamabananas $\$_1$

s_1 \$panamabanan a_6

Three Matches of ana Found!

- Searching for **ana** in panamabananas

$\$_1$panamabanana$s_1$

$a_1$bananas\$panam_1

$a_2$mabananas\$pan_1

$a_3$namabananas\$$p_1$

$a_4$nanas\$panamab_1

$a_5$nas\$panamaban_2

a_6s\$panamabanan_3

$b_1$ananas\$panama_1

$m_1$abananas\$pana_2

$n_1$amabananas\$pa_3

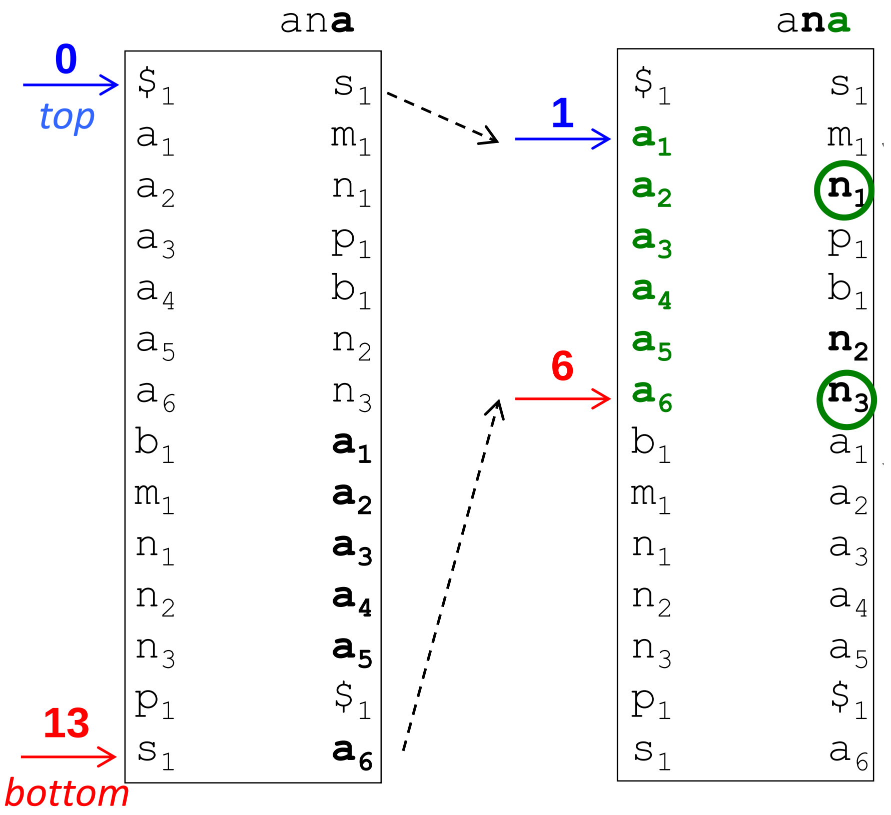
$n_2$anas\$panamaba_4

$n_3$as\$panamabana_5

$p_1$anamabananas$\$_1$

s_1\$panamabanana_6

Searching for ana using *top* and *bottom* pointers

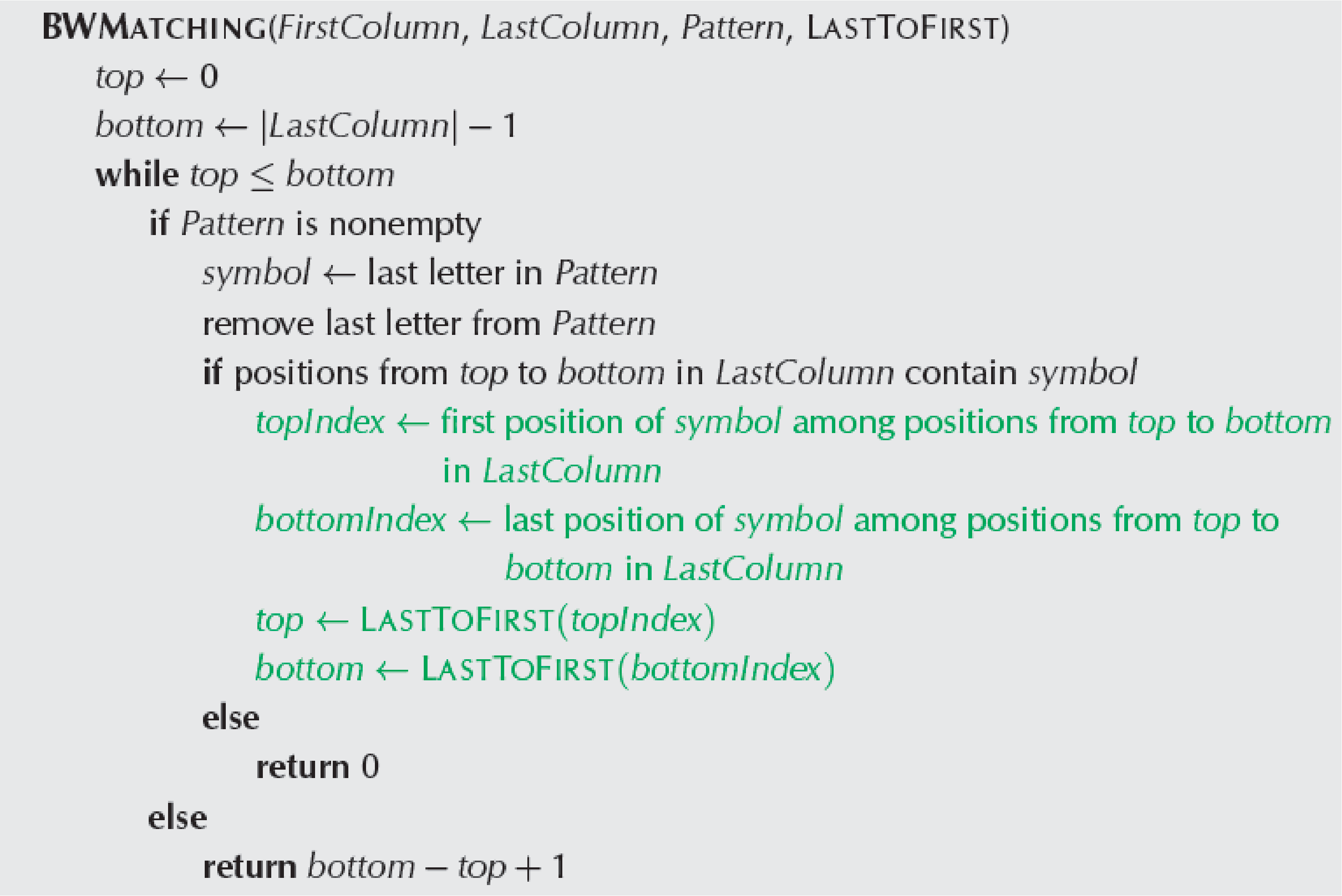


	ana
$\$_1$	s_1
a_1	m_1
a_2	n_1
a_3	p_1
a_4	b_1
a_5	n_2
a_6	n_3
b_1	$\mathbf{a_1}$
m_1	$\mathbf{a_2}$
n_1	$\mathbf{a_3}$
n_2	$\mathbf{a_4}$
n_3	$\mathbf{a_5}$
p_1	$\$_1$
s_1	$\mathbf{a_6}$

	ana
$\$_1$	s_1
$\mathbf{a_1}$	m_1
$\mathbf{a_2}$	$\mathbf{n_1}$
$\mathbf{a_3}$	p_1
$\mathbf{a_4}$	b_1
$\mathbf{a_5}$	$\mathbf{n_2}$
$\mathbf{a_6}$	$\mathbf{n_3}$
b_1	a_1
m_1	a_2
n_1	a_3
n_2	a_4
n_3	a_5
p_1	$\$_1$
s_1	a_6

```
topIndex ← first position of symbol among positions from top to bottom
        in LastColumn
bottomIndex ← last position of symbol among positions from top to
        bottom in LastColumn
```

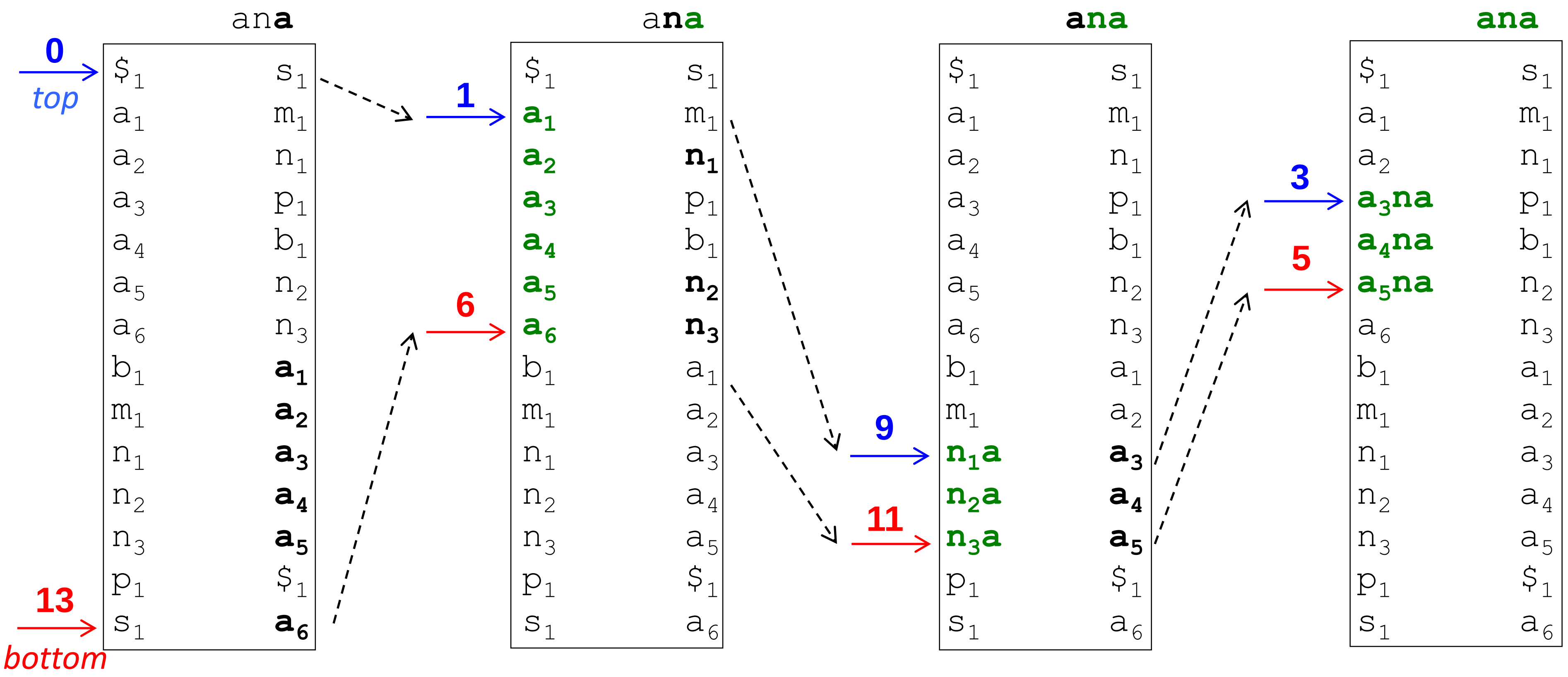
BWMatching



```
BWMATCHING(FirstColumn, LastColumn, Pattern, LASTTOFIRST)
    $top \leftarrow 0$
    $bottom \leftarrow |LastColumn| - 1$
    while $top \leq bottom$
        if Pattern is nonempty
            $symbol \leftarrow$ last letter in Pattern
            remove last letter from Pattern
            if positions from top to bottom in LastColumn contain symbol
                $topIndex \leftarrow$ first position of symbol among positions from top to bottom
                                in LastColumn
                $bottomIndex \leftarrow$ last position of symbol among positions from top to
                                bottom in LastColumn
                $top \leftarrow$ LASTTOFIRST(topIndex)
                $bottom \leftarrow$ LASTTOFIRST(bottomIndex)
            else
                return 0
        else
            return $bottom - top + 1$
```

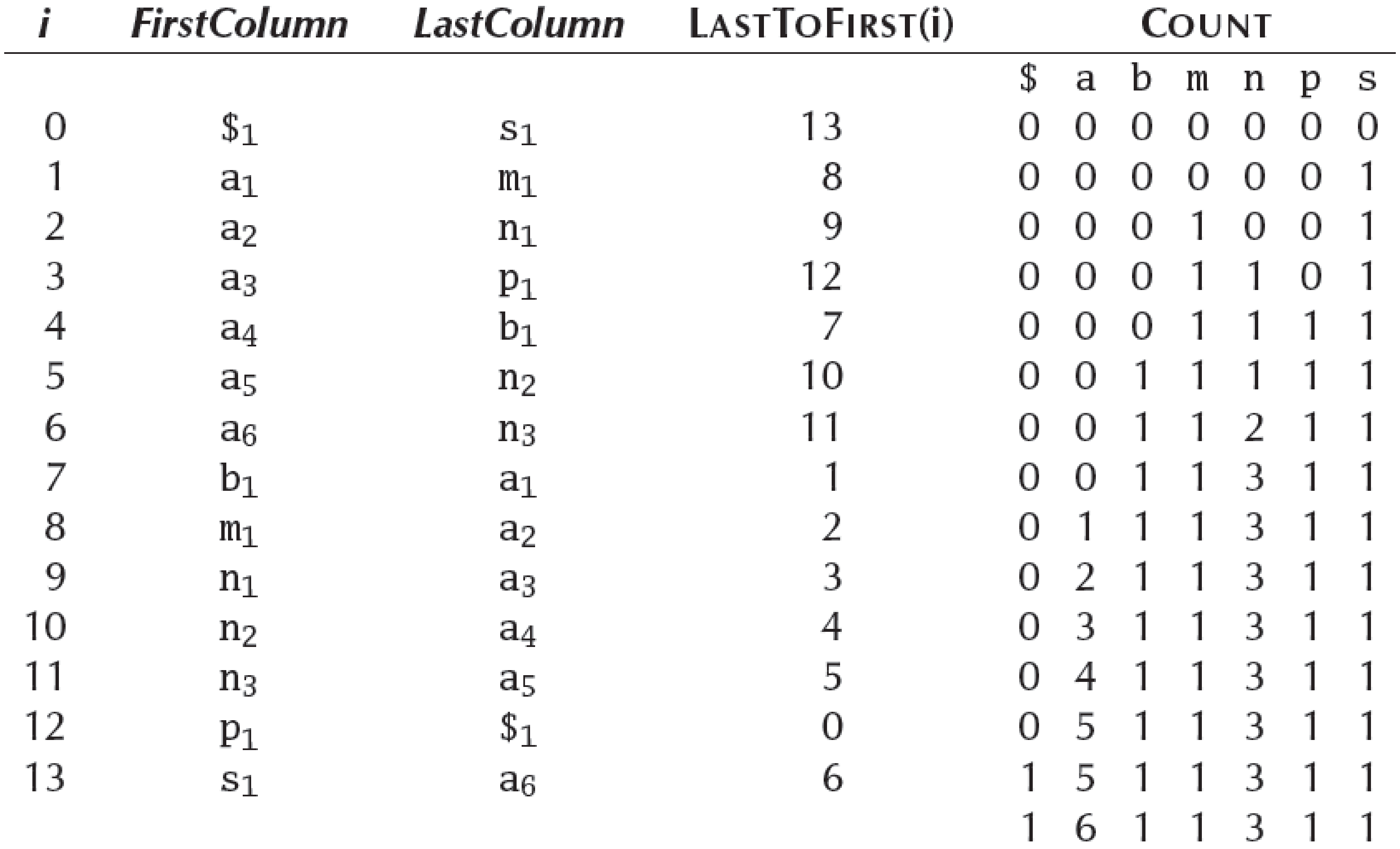
Given a symbol at position *index* in *LastColumn,*
LastToFirst(*index*) defines the position of this symbol in *FirstColumn*

BWMatching is slow: it analyzes every symbol from *top* to *bottom* in each step!



```
if positions from top to bottom in LastColumn contain symbol
    topIndex ← first position of symbol among positions from top to bottom
                in LastColumn
    bottomIndex ← last position of symbol among positions from top to
                bottom in LastColumn
```

Introducing *Count* Array



i	*FirstColumn*	*LastColumn*	LastToFirst(i)	Count						
				\$	a	b	m	n	p	s
0	$\$_1$	s_1	13	0	0	0	0	0	0	0
1	a_1	m_1	8	0	0	0	0	0	0	1
2	a_2	n_1	9	0	0	0	1	0	0	1
3	a_3	p_1	12	0	0	0	1	1	0	1
4	a_4	b_1	7	0	0	0	1	1	1	1
5	a_5	n_2	10	0	0	1	1	1	1	1
6	a_6	n_3	11	0	0	1	1	2	1	1
7	b_1	a_1	1	0	0	1	1	3	1	1
8	m_1	a_2	2	0	1	1	1	3	1	1
9	n_1	a_3	3	0	2	1	1	3	1	1
10	n_2	a_4	4	0	3	1	1	3	1	1
11	n_3	a_5	5	0	4	1	1	3	1	1
12	p_1	$\$_1$	0	0	5	1	1	3	1	1
13	s_1	a_6	6	1	5	1	1	3	1	1
				1	6	1	1	3	1	1

$\text{Count}_{symbol}(i, \textit{LastColumn})$:

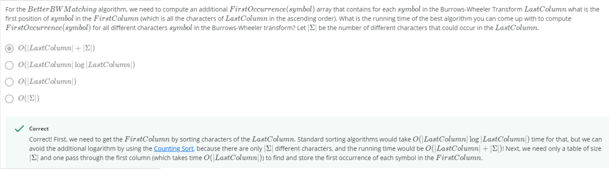
#occurrences of *symbol* in the first *i* positions of *LastColumn*

BetterBWMatching

This time complexity was very crucial to pass the test

```
BETTERBWMATCHING(FIRSTOCCURRENCE, LastColumn, Pattern, COUNT)
    top ← 0
    bottom ← |LastColumn| − 1
    while top ≤ bottom
        if Pattern is nonempty
            symbol ← last letter in Pattern
            remove last letter from Pattern
            top ← FIRSTOCCURRENCE(symbol) + COUNT_symbol(top, LastColumn)
            bottom ← FIRSTOCCURRENCE(symbol) + COUNT_symbol(bottom + 1,
                          LastColumn) − 1
        else
            return bottom − top + 1
    return
```

Notice that this code assumes that the exact pattern ("ana" here) does exist in the text



For the $BetterBWMatching$ algorithm, we need to compute an additional $FirstOccurrence(symbol)$ array that contains for each $symbol$ in the Burrows-Wheeler Transform $LastColumn$ what is the first position of $symbol$ in the $FirstColumn$ (which is all the characters of $LastColumn$ in the ascending order). What is the running time of the best algorithm you can come up with to compute $FirstOccurrence(symbol)$ for all different characters $symbol$ in the Burrows-Wheeler transform? Let $|\Sigma|$ be the number of different characters that could occur in the $LastColumn$.

● $O(|LastColumn| + |\Sigma|)$

○ $O(|LastColumn| \log |LastColumn|)$

○ $O(|LastColumn|)$

○ $O(|\Sigma|)$

✓ Correct

Correct! First, we need to get the $FirstColumn$ by sorting characters of the $LastColumn$. Standard sorting algorithms would take $O(|LastColumn| \log |LastColumn|)$ time for that, but we can avoid the additional logarithm by using the Counting Sort, because there are only $|\Sigma|$ different characters, and the running time would be $O(|LastColumn| + |\Sigma|)$! Next, we need only a table of size $|\Sigma|$ and one pass through the first column (which takes time $O(|LastColumn|)$) to find and store the first occurrence of each symbol in the $FirstColumn$.

BIOINFORMATICS ALGORITHMS
An Active Learning Approach
by Phillip Compeau & Pavel Pevzner

www.bioinformaticsalgorithms.org

Where Are the Matches?

- We know that **ana** occurs 3 times, but where does **ana** appear in *Text*???

$\$_1$panamabanana$s_1$

$a_1$bananas\$panam$_1$

$a_2$mabananas\$pan$_1$

a$_3$**na**mabananas\$p$_1$

a$_4$**na**nas\$panamab$_1$

a$_5$**na**s\$panamaban$_2$

a_6s\$panamabanan$_3$

$b_1$ananas\$panama$_1$

$m_1$abananas\$pana$_2$

$n_1$amabananas\$pa$_3$

$n_2$anas\$panamaba$_4$

$n_3$as\$panamabana$_5$

$p_1$anamabananas$\$_1$

s_1\$panamabanana$_6$

Outline

- Burrows-Wheeler Transform
- Inverting Burrows-Wheeler Transform
- Using BWT for Pattern Matching
- **Suffix Arrays**
- Approximate Pattern Matching

Where Are the Matches?

- **Suffix array** holds starting position of each suffix

$\$_1\text{panamabanana}s_1$

$a_1\text{bananas\$pana}m_1$

$a_2\text{mabananas\$pa}n_1$

$a_3\text{namabananas\$}p_1$

$a_4\text{nanas\$panama}b_1$

$a_5\text{nas\$panamaba}n_2$

$a_6\text{s\$panamabana}n_3$

$b_1\text{ananas\$panam}a_1$

$m_1\text{abananas\$pan}a_2$

$n_1\text{amabananas\$p}a_3$

$n_2\text{anas\$panamab}a_4$

$n_3\text{as\$panamaban}a_5$

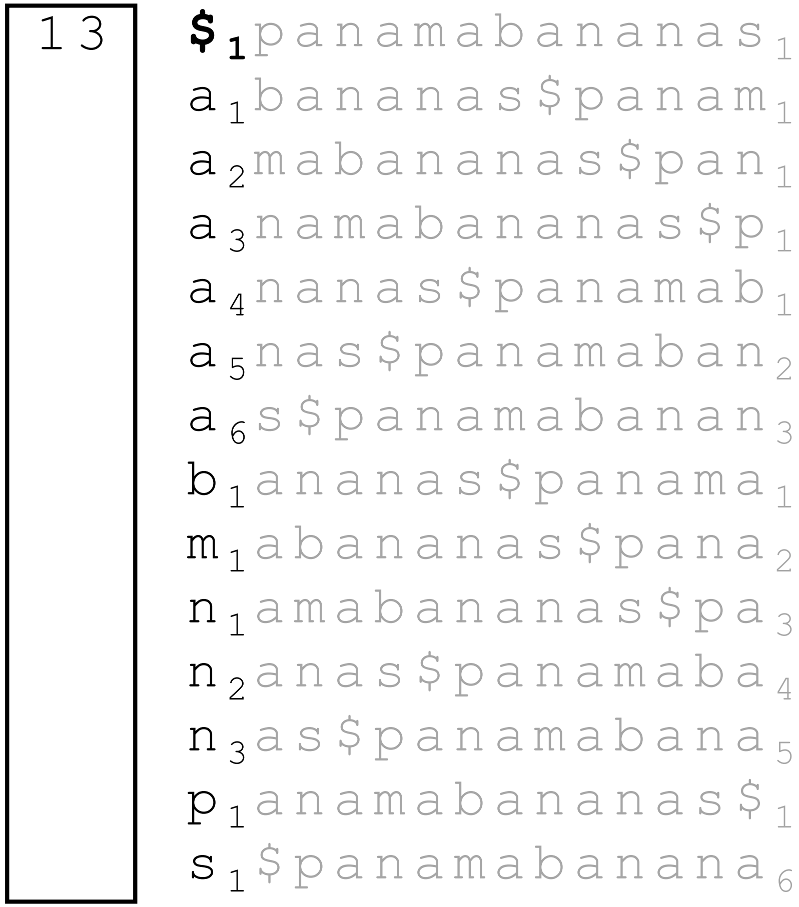
$p_1\text{anamabananas}\$_1$

$s_1\text{\$panamabanan}a_6$

Suffix Array

- **Suffix array:** holds starting position of each suffix beginning a row.

panamabananas**$**



13

**$\$_1$**panamabananas$_1$
a$_1$bananas\$panam$_1$
a$_2$mabananas\$pan$_1$
a$_3$namabananas\$p$_1$
a$_4$nanas\$panamab$_1$
a$_5$nas\$panamaban$_2$
a$_6$s\$panamabanan$_3$
b$_1$ananas\$panama$_1$
m$_1$abananas\$pana$_2$
n$_1$amabananas\$pa$_3$
n$_2$anas\$panamaba$_4$
n$_3$as\$panamabana$_5$
p$_1$anamabananas\$$_1$
s$_1$\$panamabanana$_6$

Suffix Array

- **Suffix array:** holds starting position of each suffix beginning
  a row.

panam**abananas$**

13	$\mathbf{\$_1}\mathtt{panamabananas}_1$
5	$\mathbf{a_1bananas\$}\mathtt{panam}_1$
	$\mathtt{a}_2\mathtt{mabananas\$pan}_1$
	$\mathtt{a}_3\mathtt{namabananas\$p}_1$
	$\mathtt{a}_4\mathtt{nanas\$panamab}_1$
	$\mathtt{a}_5\mathtt{nas\$panamaban}_2$
	$\mathtt{a}_6\mathtt{s\$panamabanan}_3$
	$\mathtt{b}_1\mathtt{ananas\$panama}_1$
	$\mathtt{m}_1\mathtt{abananas\$pana}_2$
	$\mathtt{n}_1\mathtt{amabananas\$pa}_3$
	$\mathtt{n}_2\mathtt{anas\$panamaba}_4$
	$\mathtt{n}_3\mathtt{as\$panamabana}_5$
	$\mathtt{p}_1\mathtt{anamabananas\$}_1$
	$\mathtt{s}_1\mathtt{\$panamabanana}_6$

Suffix Array

- **Suffix array:** holds starting position of each suffix beginning
a row.

pan**amabananas$**

13	$\mathbf{\$_1}\text{panamabananas}_1$
5	$\mathbf{a_1bananas\$}\text{panam}_1$
3	$\mathbf{a_2mabananas\$}\text{pan}_1$
	$\text{a}_3\text{namabananas\$p}_1$
	$\text{a}_4\text{nanas\$panamab}_1$
	$\text{a}_5\text{nas\$panamaban}_2$
	$\text{a}_6\text{s\$panamabanan}_3$
	$\text{b}_1\text{ananas\$panama}_1$
	$\text{m}_1\text{abananas\$pana}_2$
	$\text{n}_1\text{amabananas\$pa}_3$
	$\text{n}_2\text{anas\$panamaba}_4$
	$\text{n}_3\text{as\$panamabana}_5$
	$\text{p}_1\text{anamabananas\$}_1$
	$\text{s}_1\text{\$panamabanana}_6$

Suffix Array

- **Suffix array:** holds starting position of each suffix beginning
  a row.

p**anamabananas$**

13	**$\$_1$**panamabananas$_1$
5	**a$_1$bananas$**panam$_1$
3	**a$_2$mabananas$**pan$_1$
1	**a$_3$namabananas$**p$_1$
	a$_4$nanas$panamab$_1$
	a$_5$nas$panamaban$_2$
	a$_6$s$panamabanan$_3$
	b$_1$ananas$panama$_1$
	m$_1$abananas$pana$_2$
	n$_1$amabananaspa_3$
	n$_2$anas$panamaba$_4$
	n$_3$as$panamabana$_5$
	p$_1$anamabananas$\$_1$
	s$_1$$panamabanana$_6$

Suffix Array

- **Suffix array:** holds starting position of each suffix beginning
  a row.

13	$\mathtt{\$_1panamabananas_1}$
5	$\mathtt{a_1bananas\$panam_1}$
3	$\mathtt{a_2mabananas\$pan_1}$
1	$\mathtt{a_3namabananas\$p_1}$
7	$\mathtt{a_4nanas\$panamab_1}$
	$\mathtt{a_5nas\$panamaban_2}$
	$\mathtt{a_6s\$panamabanan_3}$
	$\mathtt{b_1ananas\$panama_1}$
	$\mathtt{m_1abananas\$pana_2}$
	$\mathtt{n_1amabananas\$pa_3}$
	$\mathtt{n_2anas\$panamaba_4}$
	$\mathtt{n_3as\$panamabana_5}$
	$\mathtt{p_1anamabananas\$_1}$
	$\mathtt{s_1\$panamabanana_6}$

panamab**ananas$**

Suffix Array

- **Suffix array:** holds starting position of each suffix beginning
  a row.

13	**$\$_1$**panamabananas$_1$
5	**a$_1$bananas\$**panam$_1$
3	**a$_2$mabananas\$**pan$_1$
1	**a$_3$namabananas\$**p$_1$
7	**a$_4$nanas\$**panamab$_1$
9	**a$_5$nas\$**panamaban$_2$
	a$_6$s\$panamabanan$_3$
	b$_1$ananas\$panama$_1$
	m$_1$abananas\$pana$_2$
	n$_1$amabananas\$pa$_3$
	n$_2$anas\$panamaba$_4$
	n$_3$as\$panamabana$_5$
	p$_1$anamabananas\$$_1$
	s$_1$\$panamabanana$_6$

panamaban**anas$**

Suffix Array

- **Suffix array:** holds starting position of each suffix beginning
  a row.

panamabanan**as\$**

13	**\$$_1$**panamabananas$_1$
5	**a$_1$bananas\$**panam$_1$
3	**a$_2$mabananas\$**pan$_1$
1	**a$_3$namabananas\$**p$_1$
7	**a$_4$nanas\$**panamab$_1$
9	**a$_5$nas\$**panamaban$_2$
11	**a$_6$s\$**panamabanan$_3$
	b$_1$ananas\$panama$_1$
	m$_1$abananas\$pana$_2$
	n$_1$amabananas\$pa$_3$
	n$_2$anas\$panamaba$_4$
	n$_3$as\$panamabana$_5$
	p$_1$anamabananas\$$_1$
	s$_1$\$panamabanana$_6$

Suffix Array

- **Suffix array:** holds starting position of each suffix beginning a row.

panama**bananas$**

13	$\mathbf{\$_1}\text{panamabananas}_1$
5	$\mathbf{a_1bananas\$}\text{panam}_1$
3	$\mathbf{a_2mabananas\$}\text{pan}_1$
1	$\mathbf{a_3namabananas\$}\text{p}_1$
7	$\mathbf{a_4nanas\$}\text{panamab}_1$
9	$\mathbf{a_5nas\$}\text{panamaban}_2$
11	$\mathbf{a_6s\$}\text{panamabanan}_3$
6	$\mathbf{b_1ananas\$}\text{panama}_1$
	$m_1\text{abananas\$pana}_2$
	$n_1\text{amabananas\$pa}_3$
	$n_2\text{anas\$panamaba}_4$
	$n_3\text{as\$panamabana}_5$
	$p_1\text{anamabananas\$}_1$
	$s_1\text{\$panamabanana}_6$

Suffix Array

- **Suffix array:** holds starting position of each suffix beginning
  a row.

pana**mabananas$**

13	$\texttt{\$}_1\texttt{panamabananas}_1$
5	$\texttt{a}_1\texttt{bananas\$panam}_1$
3	$\texttt{a}_2\texttt{mabananas\$pan}_1$
1	$\texttt{a}_3\texttt{namabananas\$p}_1$
7	$\texttt{a}_4\texttt{nanas\$panamab}_1$
9	$\texttt{a}_5\texttt{nas\$panamaban}_2$
11	$\texttt{a}_6\texttt{s\$panamabanan}_3$
6	$\texttt{b}_1\texttt{ananas\$panama}_1$
4	$\texttt{m}_1\texttt{abananas\$pana}_2$
	$\texttt{n}_1\texttt{amabananas\$pa}_3$
	$\texttt{n}_2\texttt{anas\$panamaba}_4$
	$\texttt{n}_3\texttt{as\$panamabana}_5$
	$\texttt{p}_1\texttt{anamabananas\$}_1$
	$\texttt{s}_1\texttt{\$panamabanana}_6$

Suffix Array

- **Suffix array:** holds starting position of each suffix beginning
  a row.

13	**$\$_1$**panamabananas$_1$
5	**a$_1$bananas$\$$**panam$_1$
3	**a$_2$mabananas$\$$**pan$_1$
1	**a$_3$namabananas$\$$**p$_1$
7	**a$_4$nanas$\$$**panamab$_1$
9	**a$_5$nas$\$$**panamaban$_2$
11	**a$_6$s$\$$**panamabanan$_3$
6	**b$_1$ananas$\$$**panama$_1$
4	**m$_1$abananas$\$$**pana$_2$
2	**n$_1$amabananas$\$$**pa$_3$
	n$_2$anas$\$$panamaba$_4$
	n$_3$as$\$$panamabana$_5$
	p$_1$anamabananas$\$_1$
	s$_1$$\$$panamabanana$_6$

pa**namabananas$\$$**

Suffix Array

- **Suffix array:** holds starting position of each suffix beginning a row.

13	**\$**$_1$panamabananas$_1$
5	**a**$_1$**bananas\$**panam$_1$
3	**a**$_2$**mabananas\$**pan$_1$
1	**a**$_3$**namabananas\$**p$_1$
7	**a**$_4$**nanas\$**panamab$_1$
9	**a**$_5$**nas\$**panamaban$_2$
11	**a**$_6$**s\$**panamabanan$_3$
6	**b**$_1$**ananas\$**panama$_1$
4	**m**$_1$**abananas\$**pana$_2$
2	**n**$_1$**amabananas\$**pa$_3$
8	**n**$_2$**anas\$**panamaba$_4$
	n$_3$as\$panamabana$_5$
	p$_1$anamabananas\$$_1$
	s$_1$\$panamabanana$_6$

panamaba**nanas\$**

Suffix Array

- **Suffix array:** holds starting position of each suffix beginning a row.

13	$\mathbf{\$_1}\texttt{panamabananas}_1$
5	$\mathbf{a_1bananas\$}\texttt{panam}_1$
3	$\mathbf{a_2mabananas\$}\texttt{pan}_1$
1	$\mathbf{a_3namabananas\$}\texttt{p}_1$
7	$\mathbf{a_4nanas\$}\texttt{panamab}_1$
9	$\mathbf{a_5nas\$}\texttt{panamaban}_2$
11	$\mathbf{a_6s\$}\texttt{panamabanan}_3$
6	$\mathbf{b_1ananas\$}\texttt{panama}_1$
4	$\mathbf{m_1abananas\$}\texttt{pana}_2$
2	$\mathbf{n_1amabananas\$}\texttt{pa}_3$
8	$\mathbf{n_2anas\$}\texttt{panamaba}_4$
10	$\mathbf{n_3as\$}\texttt{panamabana}_5$
	$\texttt{p}_1\texttt{anamabananas\$}_1$
	$\texttt{s}_1\texttt{\$panamabanana}_6$

panamabana**nas$**

Suffix Array

- **Suffix array:** holds starting position of each suffix beginning
  a row.

panamabananas$

13	$\mathbf{\$}_1\text{panamabananas}_1$
5	$\mathbf{a}_1\mathbf{bananas\$}\text{panam}_1$
3	$\mathbf{a}_2\mathbf{mabananas\$}\text{pan}_1$
1	$\mathbf{a}_3\mathbf{namabananas\$}\text{p}_1$
7	$\mathbf{a}_4\mathbf{nanas\$}\text{panamab}_1$
9	$\mathbf{a}_5\mathbf{nas\$}\text{panamaban}_2$
11	$\mathbf{a}_6\mathbf{s\$}\text{panamabanan}_3$
6	$\mathbf{b}_1\mathbf{ananas\$}\text{panama}_1$
4	$\mathbf{m}_1\mathbf{abananas\$}\text{pana}_2$
2	$\mathbf{n}_1\mathbf{amabananas\$}\text{pa}_3$
8	$\mathbf{n}_2\mathbf{anas\$}\text{panamaba}_4$
10	$\mathbf{n}_3\mathbf{as\$}\text{panamabana}_5$
0	$\mathbf{p}_1\mathbf{anamabananas\$}_1$
	$\text{s}_1\text{\$panamabanana}_6$

Suffix Array

- **Suffix array:** holds starting position of each suffix beginning
  a row.

13	**$\$_1$**panamabanana$s_1$
5	**$a_1$bananas\$**panam_1
3	**$a_2$mabananas\$**pan_1
1	**$a_3$namabananas\$**p_1
7	**$a_4$nanas\$**panamab_1
9	**$a_5$nas\$**panamaban_2
11	**a_6s\$**panamabanan_3
6	**$b_1$ananas\$**panama_1
4	**$m_1$abananas\$**pana_2
2	**$n_1$amabananas\$**pa_3
8	**$n_2$anas\$**panamaba_4
10	**$n_3$as\$**panamabana_5
0	**$p_1$anamabananas$\$_1$**
12	**s_1\$**panamabanana_6

panamabanana**s\$**

Suffix Array

- **Suffix array:** holds starting position of each suffix beginning
  a row.

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

Using the Suffix Array to Find Matches

- Thus, **ana** occurs at positions **1**, **7**, **9**:
- panamabananas$

↑ ↑ ↑

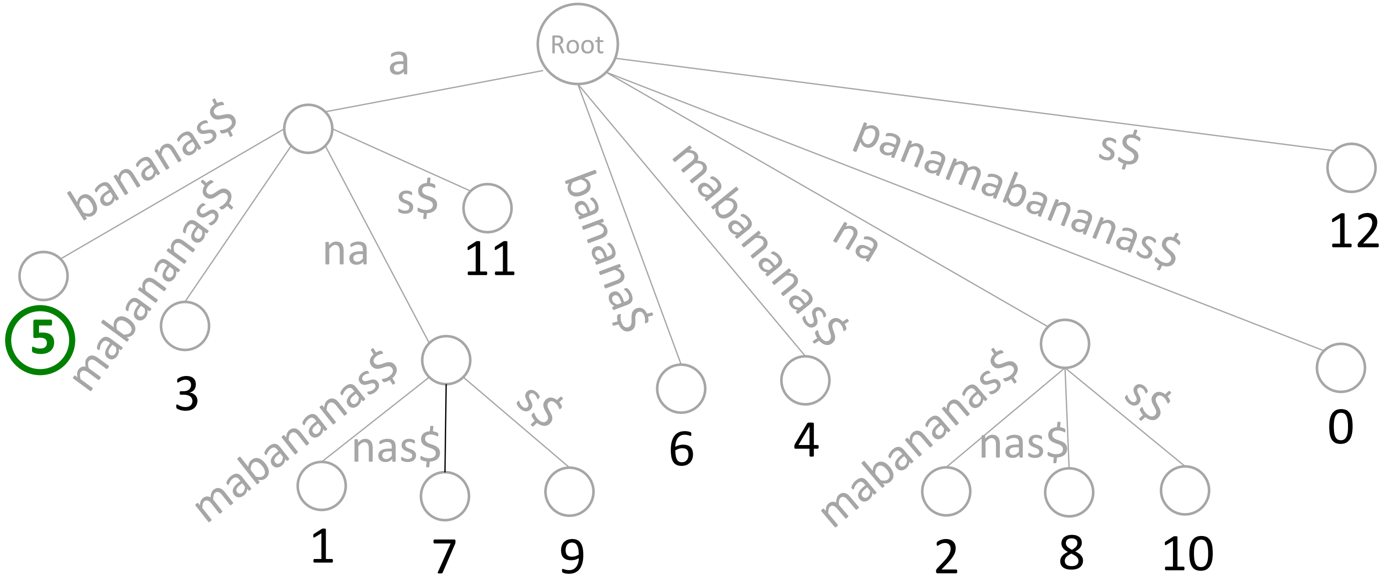
13	$\$_1$panamabanana$s_1$
5	$a_1$bananas\$panam_1
3	$a_2$mabananas\$pan_1
1	a_3**na**mabananas\$$p_1$
7	a_4**na**nas\$panamab_1
9	a_5**na**s\$panamaban_2
11	a_6s\$panamabanan_3
6	$b_1$ananas\$panama_1
4	$m_1$abananas\$pana_2
2	$n_1$amabananas\$pa_3
8	$n_2$anas\$panamaba_4
10	$n_3$as\$panamabana_5
0	$p_1$anamabananas$\$_1$
12	s_1\$panamabanana_6

Naïve algorithm for constructing suffix array (sorting all suffixes of *Text*)

O(|*Text*| • log|*Text*|) comparisons

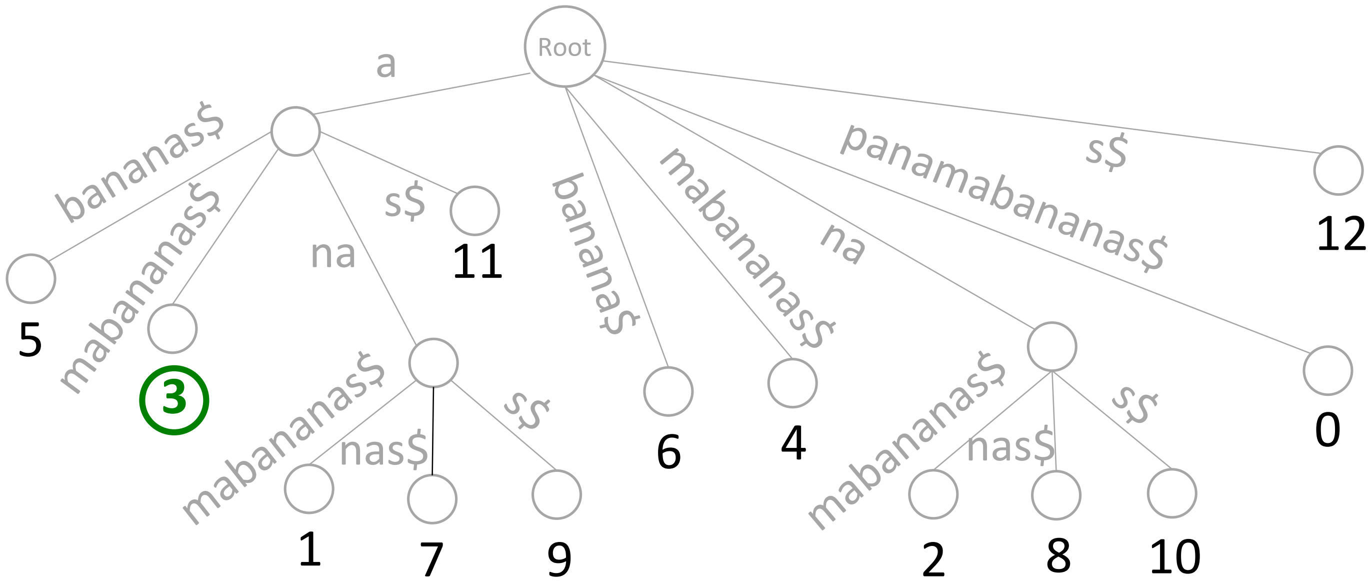
How can we quicken this. See next

From Suffix Tree to Suffix Array: Depth-First Traversal



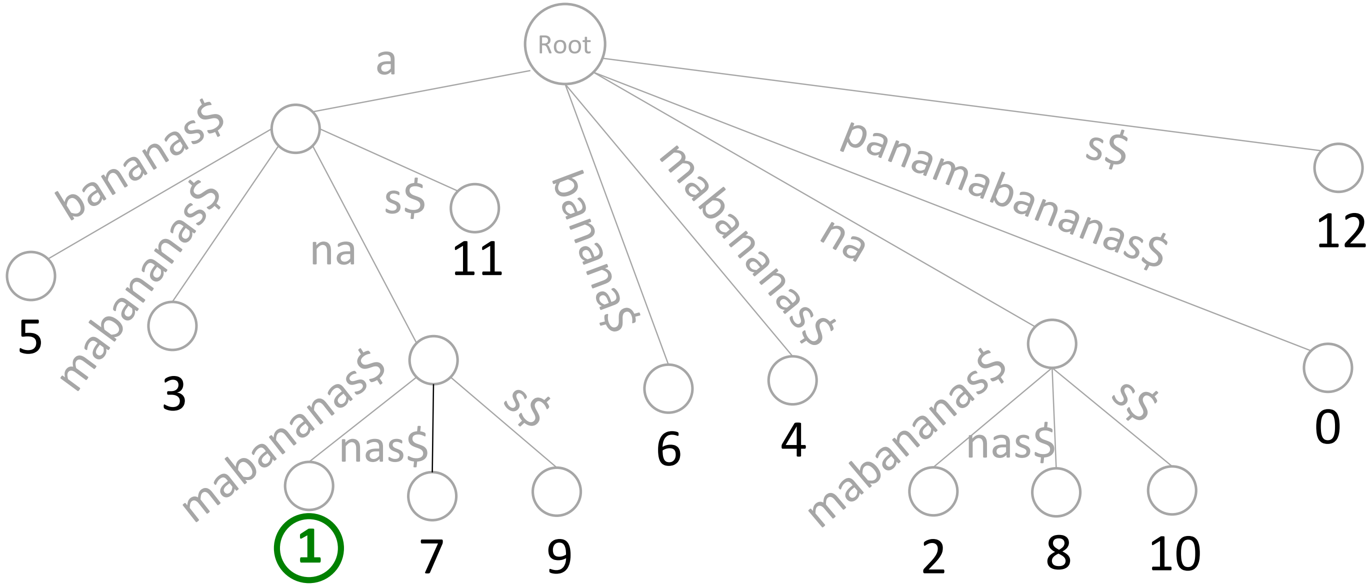
[13 5 3 1 7 9 11 6 4 2 8 10 0 12]

From Suffix Tree to Suffix Array: Depth-First Traversal



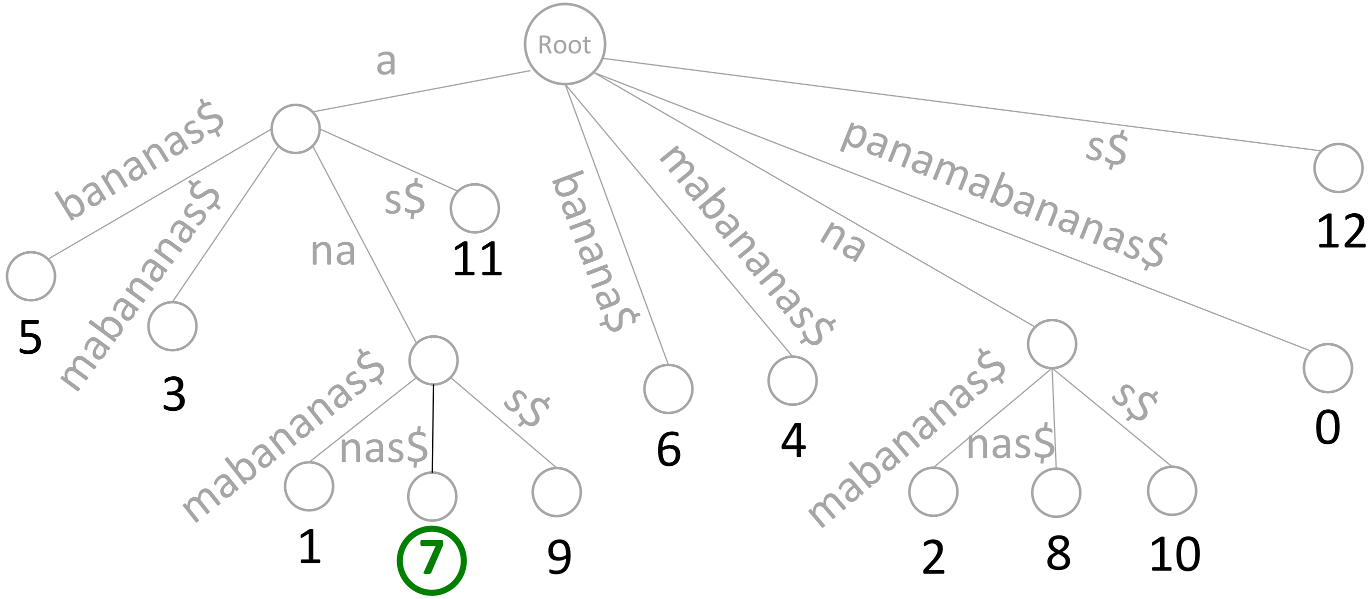
[13 5 3 1 7 9 11 6 4 2 8 10 0 12]

From Suffix Tree to Suffix Array: Depth-First Traversal



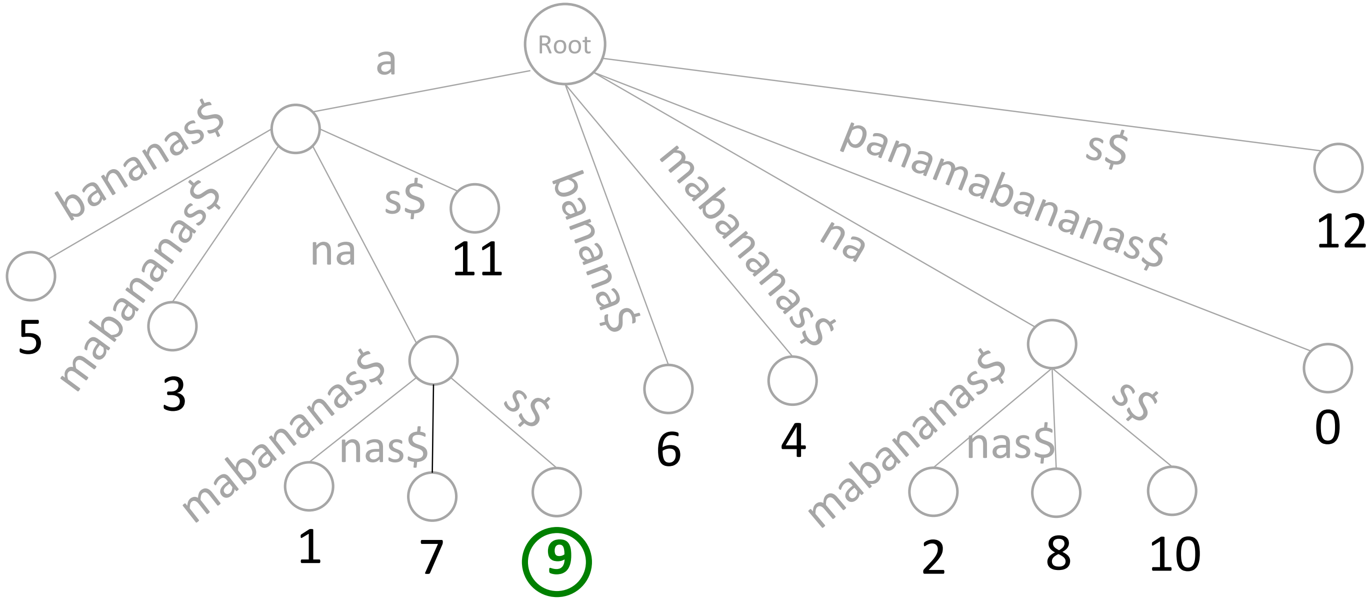
[13 5 3 1 7 9 11 6 4 2 8 10 0 12]

From Suffix Tree to Suffix Array



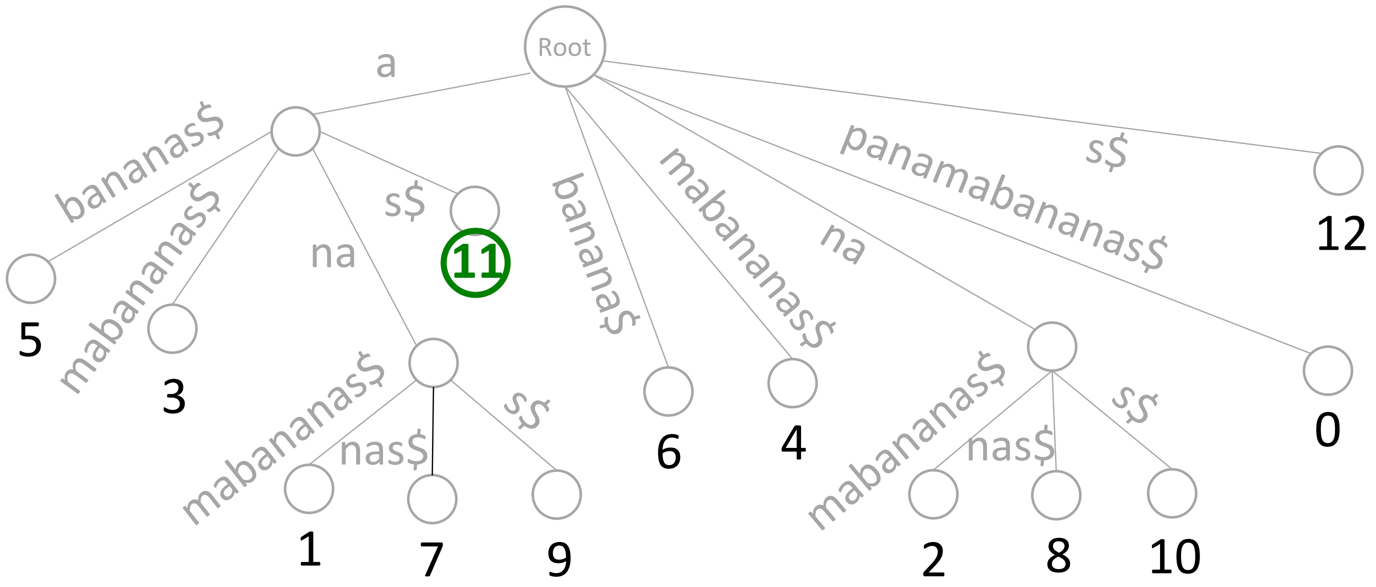
[13 5 3 1 7 9 11 6 4 2 8 10 0 12]

From Suffix Tree to Suffix Array



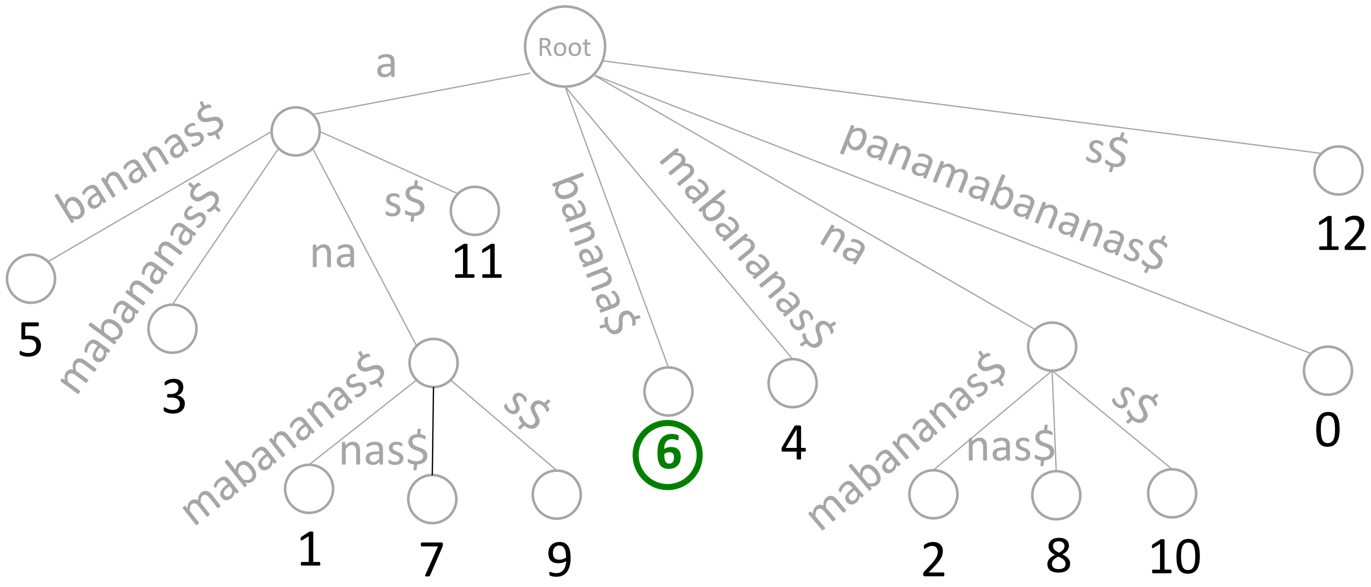
[13 5 3 1 7 9 11 6 4 2 8 10 0 12]

From Suffix Tree to Suffix Array



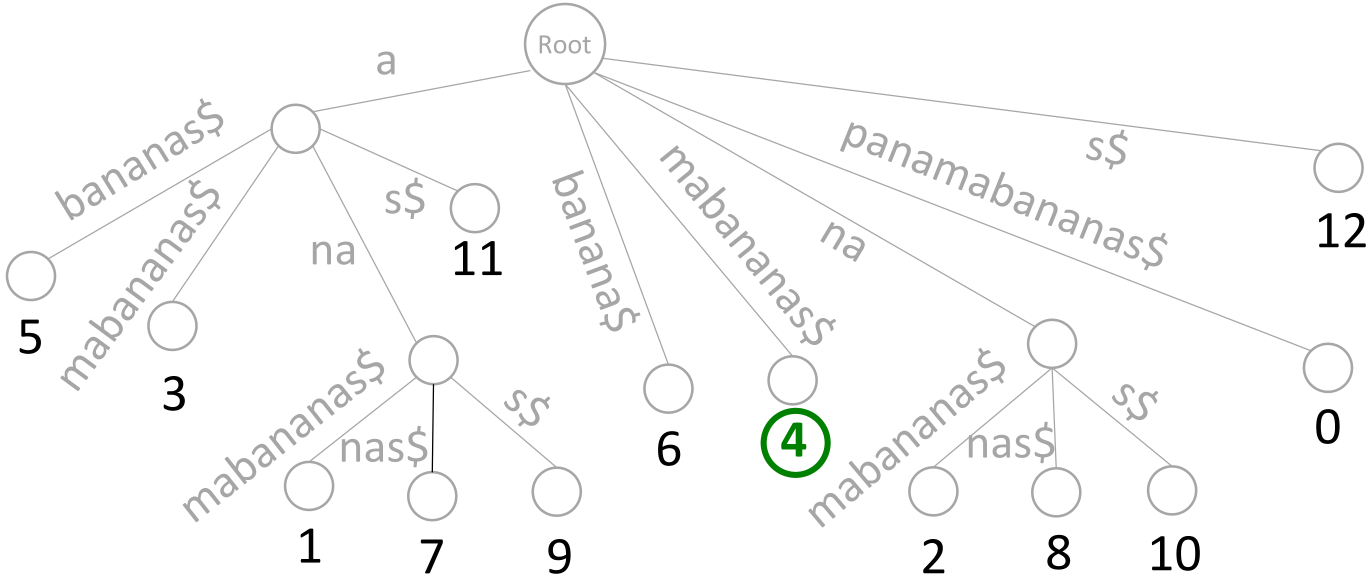
[13 5 3 1 7 9 11 6 4 2 8 10 0 12]

From Suffix Tree to Suffix Array



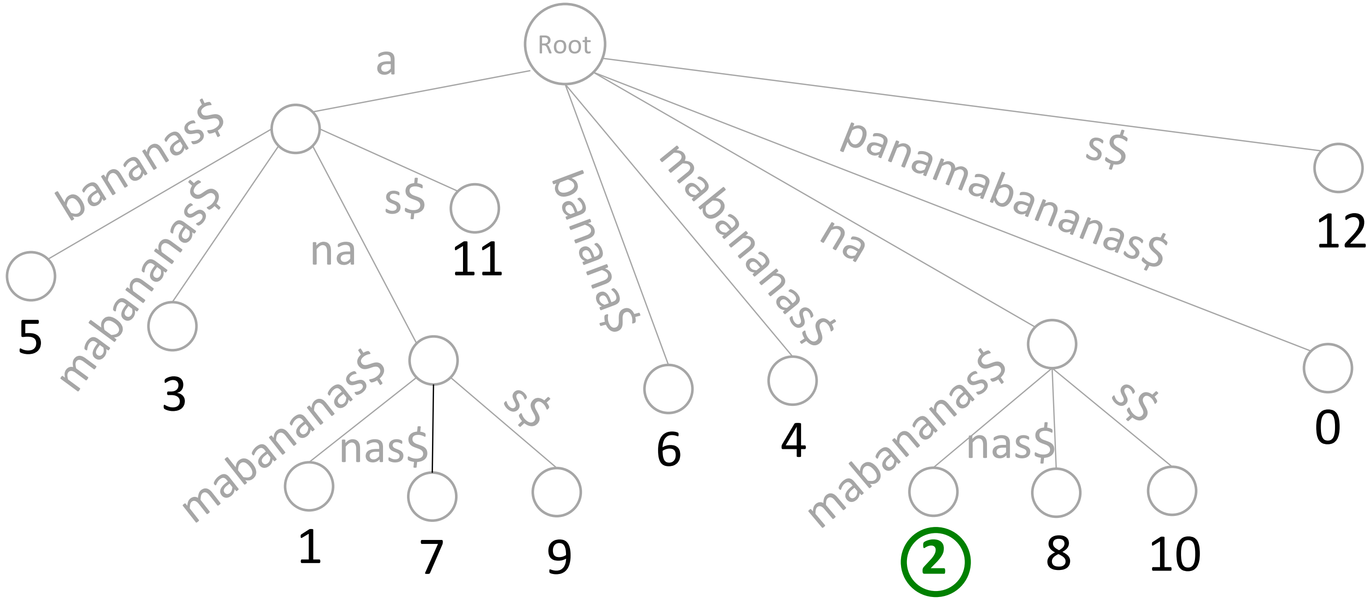
[13 5 3 1 7 9 11 6 4 2 8 10 0 12]

From Suffix Tree to Suffix Array



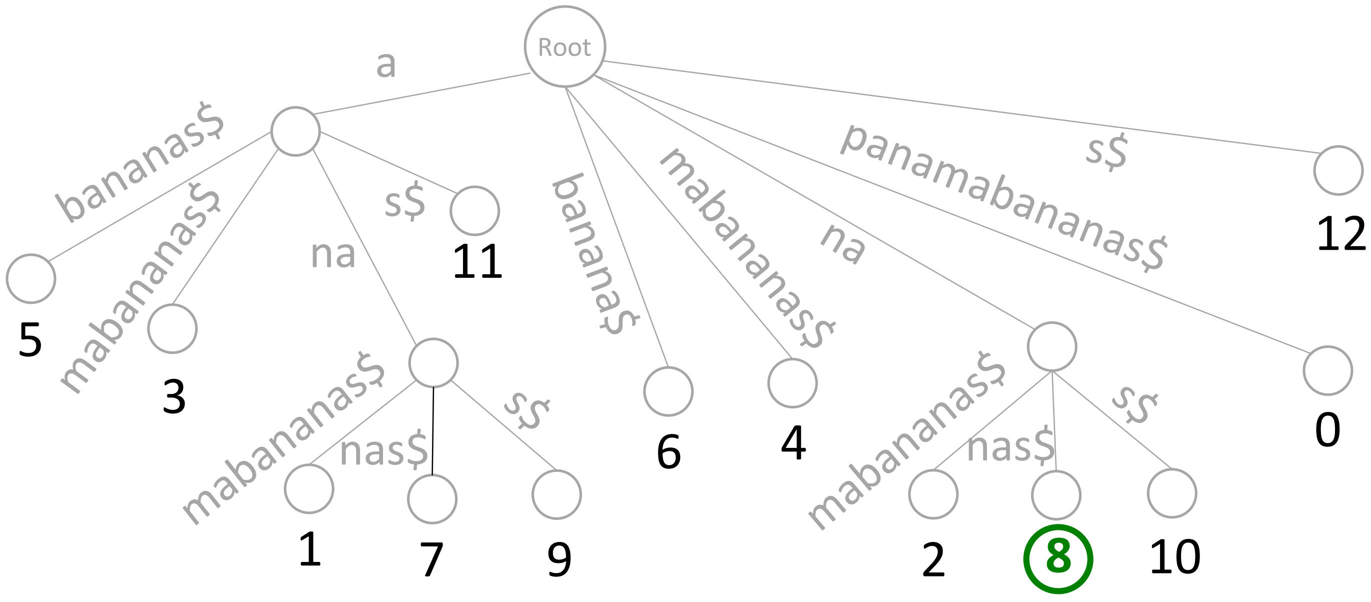
[13 5 3 1 7 9 11 6 4 2 8 10 0 12]

From Suffix Tree to Suffix Array



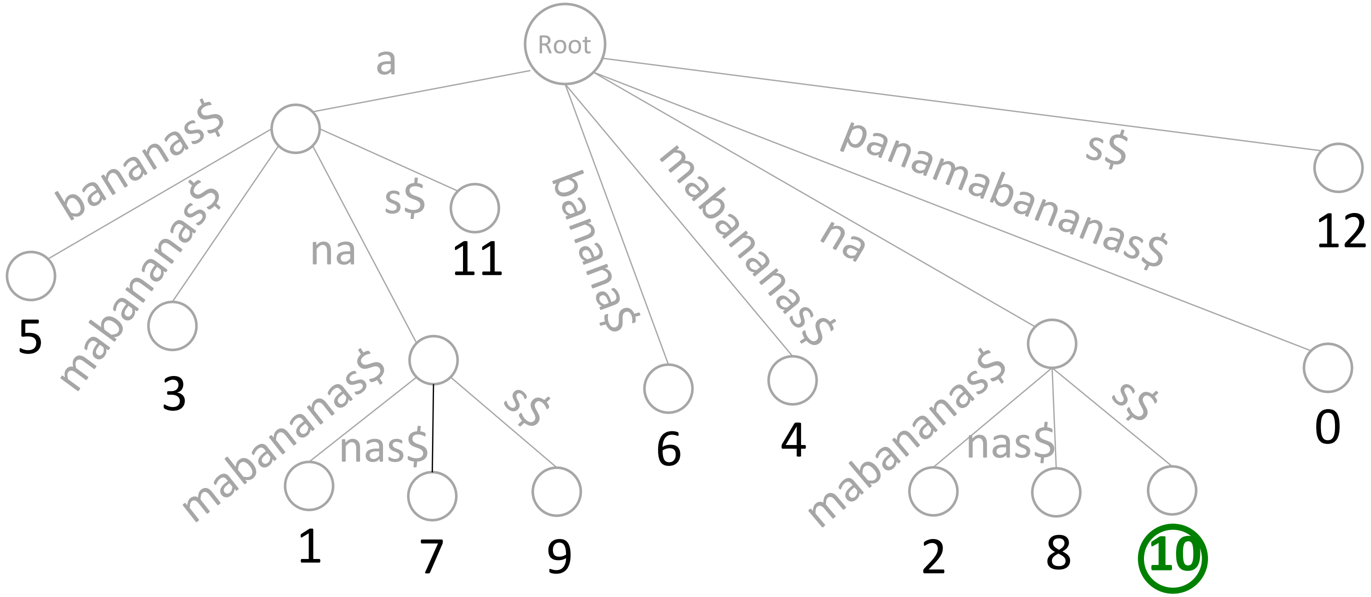
[13 5 3 1 7 9 11 6 4 2 8 10 0 12]

From Suffix Tree to Suffix Array



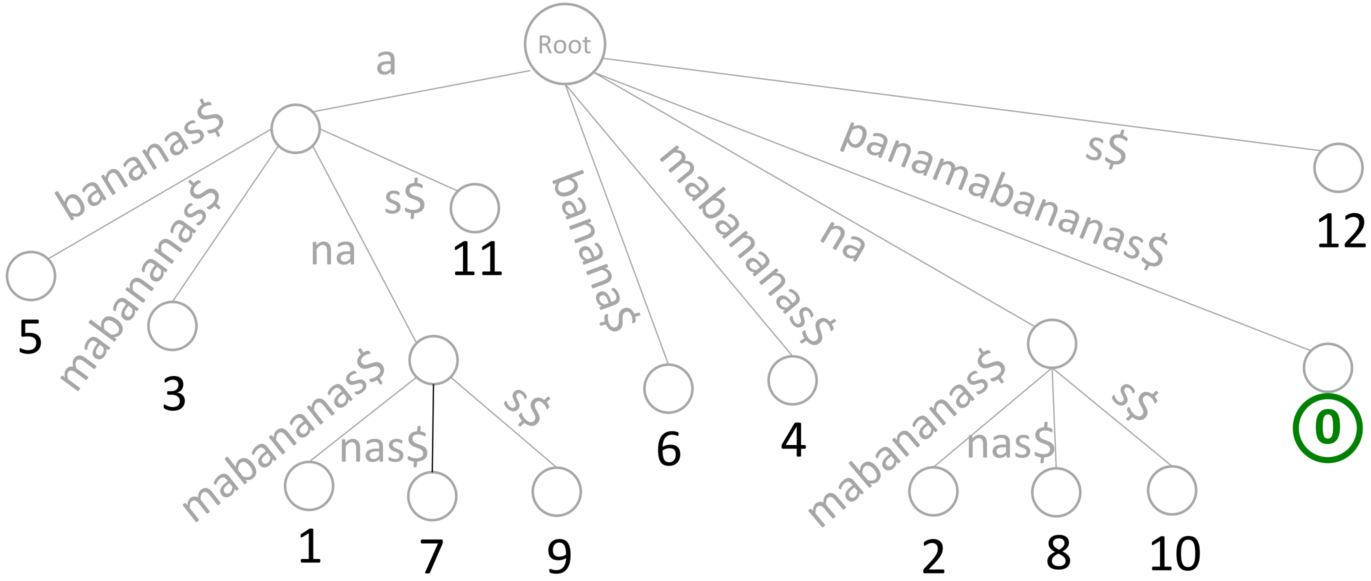
[13 5 3 1 7 9 11 6 4 2 8 10 0 12]

From Suffix Tree to Suffix Array



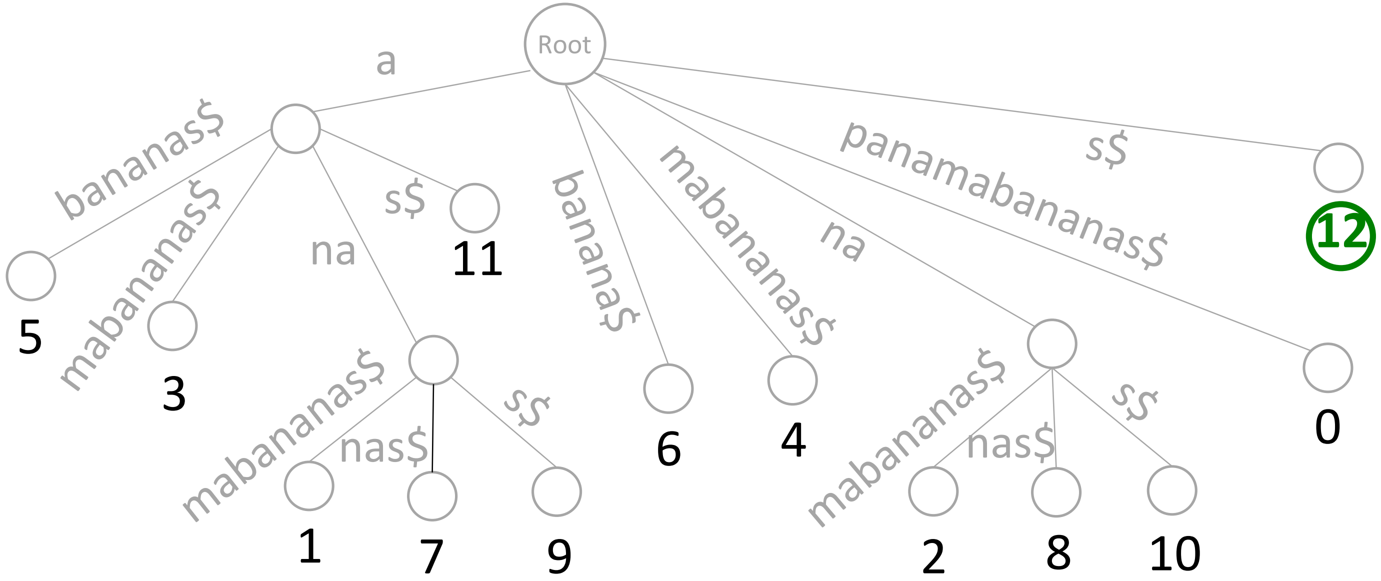
[13 5 3 1 7 9 11 6 4 2 8 10 0 12]

From Suffix Tree to Suffix Array



[13 5 3 1 7 9 11 6 4 2 8 10 0 12]

From Suffix Tree to Suffix Array



[13 5 3 1 7 9 11 6 4 2 8 10 0 12]

Constructing Suffix Array

- Depth-first traversal of suffix tree
  - O(|*Text*|) time and ~20•|*Text*| space
- Manber-Myers algorithm (1990):
  - O(|*Text*|) time and ~4•|*Text*| space
- But memory footprint is still large for human genome!

We will learn how to quickly construct suffix array without relying on suffix tree later in this course

Reducing Memory Footprint for Suffix Array

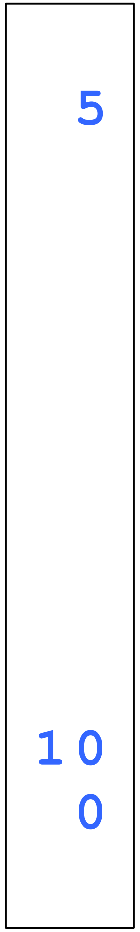
As 4|Text| is still large amount of memory

- Can we store only a fraction of the suffix array but still do fast pattern matching?

13
5
3
1
7
9
11
6
4
2
8
10
0
12

Reducing Memory Footprint for Suffix Array

- Can we store only a fraction of the suffix array but still do fast pattern matching?
- Partial suffix array SuffixArray$_K$(*Text*) only contains values that are multiples of some integer *K*



As 4|Text| is still large amount of memory

Using the Suffix Array to Find Matches

	suffix array
$\$_1$panamabanana$s_1$	13
$a_1$bananas\$panam_1	5
$a_2$mabananas\$pan_1	3
$a_3$namabananas\$$p_1$	**1**
$a_4$nanas\$panamab_1	**7**
$a_5$nas\$panamaban_2	**9**
a_6s\$panamabanan_3	11
$b_1$ananas\$panama_1	6
$m_1$abananas\$pana_2	4
$n_1$amabananas\$pa_3	2
$n_2$anas\$panamaba_4	8
$n_3$as\$panamabana_5	10
$p_1$anamabananas$\$_1$	0
s_1\$panamabanana_6	12

Using the Partial Suffix Array to Find Matches

	partial suffix array
$\$_1\text{panamabanana}s_1$	
$a_1\text{bananas\$pana}m_1$	5
$a_2\text{mabananas\$pa}n_1$	
$\mathbf{a_3na}\text{mabananas\$}p_1$	
$\mathbf{a_4na}\text{nas\$panama}b_1$	
$\mathbf{a_5na}\text{s\$panamaba}n_2$	
$a_6\text{s\$panamabana}n_3$	
$b_1\text{ananas\$panam}a_1$	
$m_1\text{abananas\$pan}a_2$	
$n_1\text{amabananas\$p}a_3$	
$n_2\text{anas\$panamab}a_4$	
$n_3\text{as\$panamaban}a_5$	10
$p_1\text{anamabananas}\$_1$	0
$s_1\text{\$panamabanan}a_6$	

Using the Partial Suffix Array to Find Matches

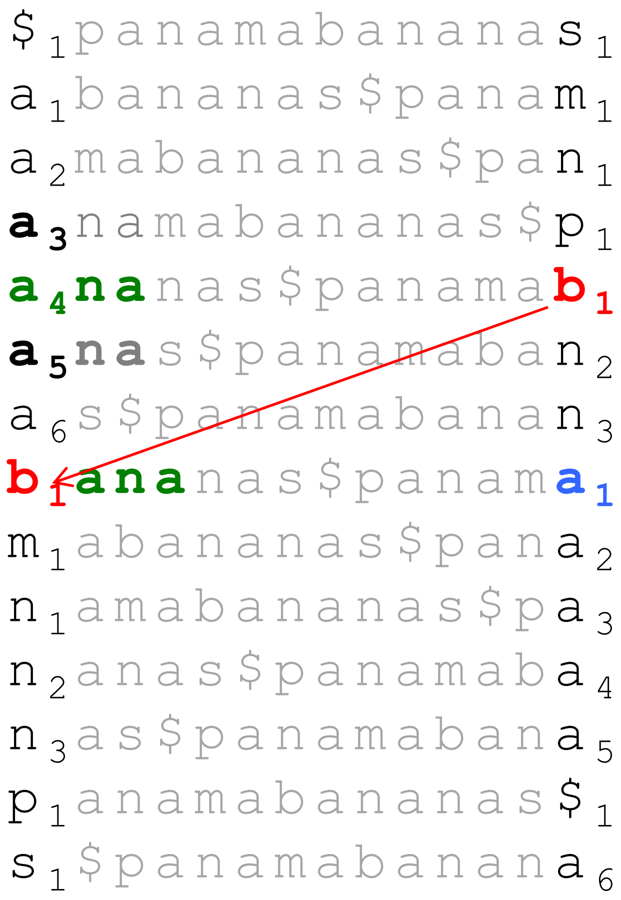
	partial suffix array
$\$_1\text{panamabanana}s_1$	
$a_1\text{bananas\$pana}m_1$	5
$a_2\text{mabananas\$pa}n_1$	
$\mathbf{a_3na}\text{mabananas\$}p_1$	
$\mathbf{a_4na}\text{nas\$panama}b_1$	
$\mathbf{a_5na}\text{s\$panamaba}n_2$	
$a_6\text{s\$panamabana}n_3$	
$b_1\text{ananas\$panam}a_1$	
$m_1\text{abananas\$pan}a_2$	
$n_1\text{amabananas\$p}a_3$	
$n_2\text{anas\$panamab}a_4$	
$n_3\text{as\$panamaban}a_5$	10
$p_1\text{anamabananas}\$_1$	0
$s_1\text{\$panamabanan}a_6$	

Where are these **ana** prefixes located in *Text*???

Focus on a_4na

		partial suffix array
$\$_1$panamabanana$s_1$		
$a_1$bananas\$panam_1		5
$a_2$mabananas\$pan_1		
$a_3$namabananas\$$p_1$		
$a_4$nanas\$panamab_1	Where is a_4na?	
$a_5$nas\$panamaban_2		
a_6s\$panamabanan_3		
$b_1$ananas\$panama_1		
$m_1$abananas\$pana_2		
$n_1$amabananas\$pa_3		
$n_2$anas\$panamaba_4		
$n_3$as\$panamabana_5		10
$p_1$anamabananas$\$_1$		0
s_1\$panamabanana_6		

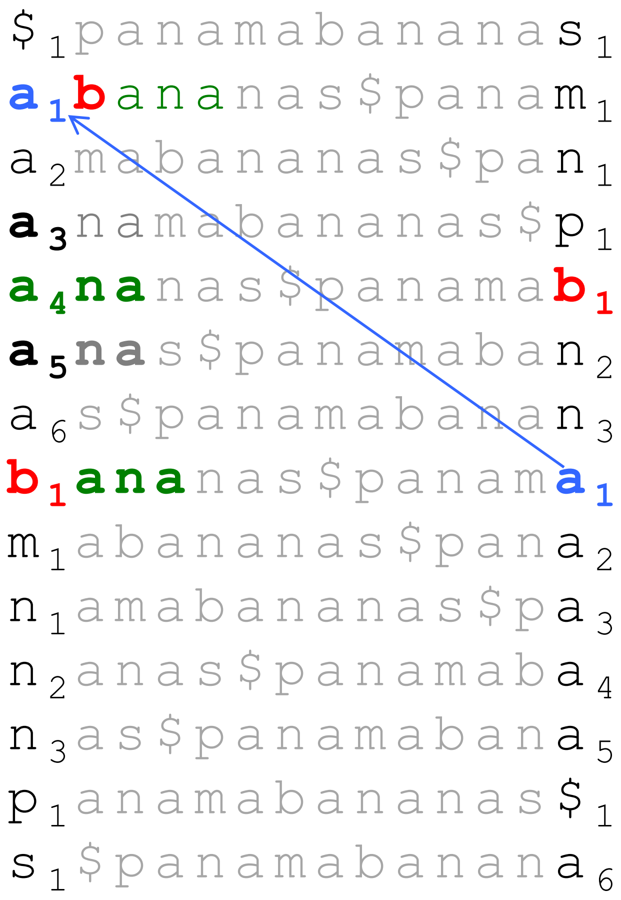
Focus on $b_1$ana



			partial suffix array
$\$_1$	panamabanana	s_1	
a_1	bananas$panam	m_1	5
a_2	mabananas$pan	n_1	
a_3	namabananas$p	p_1	
a_4	nanas$panama	b_1	
a_5	nas$panamaban	n_2	
a_6	s$panamabanan	n_3	
b_1	ananas$panam	a_1	
m_1	abananas$pana	a_2	
n_1	amabananas$pa	a_3	
n_2	anas$panamaba	a_4	
n_3	as$panamabana	a_5	10
p_1	anamabananas$	$\$_1$	0
s_1	$panamabanana	a_6	

Where is $b_1$ana?

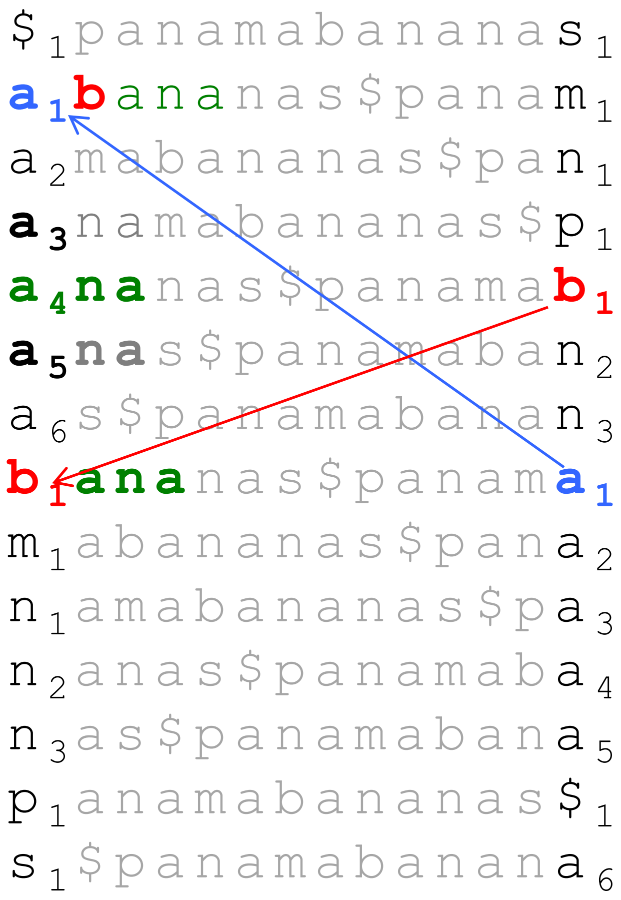
Focus on $a_1$bana



Where is $a_1$bana?

F		L	partial suffix array
$\$_1$	panamabanana	s_1	
a_1	bananas$panam	m_1	5
a_2	mabananas$pa	n_1	
a_3	namabananas$	p_1	
a_4	nanas$panama	b_1	
a_5	nas$panamaba	n_2	
a_6	s$panamabana	n_3	
b_1	ananas$panam	a_1	
m_1	abananas$pan	a_2	
n_1	amabananas$p	a_3	
n_2	anas$panamab	a_4	
n_3	as$panamaban	a_5	10
p_1	anamabananas	$\$_1$	0
s_1	$panamabanan	a_6	

Partial suffix array reveals position of $a_1$bana



			partial suffix array
`$1`	`panamabanana`	`s1`	
`a1`	`bananas$pana`	`m1`	5
`a2`	`mabananas$pa`	`n1`	
`a3`	`namabananas$`	`p1`	
`a4`	`nanas$panama`	`b1`	7
`a5`	`nas$panamaba`	`n2`	
`a6`	`s$panamabana`	`n3`	
`b1`	`ananas$panam`	`a1`	6
`m1`	`abananas$pan`	`a2`	
`n1`	`amabananas$p`	`a3`	
`n2`	`anas$panamab`	`a4`	
`n3`	`as$panamaban`	`a5`	10
`p1`	`anamabananas`	`$1`	0
`s1`	`$panamabanan`	`a6`	

$a_1$bana is at position 5

$a_4$na is at position 7

$b_1$ana is at position 6

Outline

- Burrows-Wheeler Transform
- Inverting Burrows-Wheeler Transform
- Using BWT for Pattern Matching
- Suffix Arrays
- **Approximate Pattern Matching**

Returning to Search for Mutations

- **Approximate Pattern Matching Problem**:
    - **Input**: A string *Pattern*, a string *Text*, and an integer *d*.
    - **Output:** All positions in *Text* where the string *Pattern* appears as a substring with at most *d* mismatches.

Revealing Mutations by Analyzing Billions of Reads

- **Multiple Approximate Pattern Matching Problem**
  - **Input**: A **set** of strings *Patterns*, a string *Text*, and an integer *d*.
  - **Output:** All positions in *Text* where a string from *Patterns* appears as a substring with at most *d* mismatches.

BWT Saves the Day Again

- searching for `ana` in `panamabananas`

$\$_1\texttt{panamabanana}s_1$

$a_1\texttt{bananas\$pana}m_1$

$a_2\texttt{mabananas\$pa}n_1$

$a_3\texttt{namabananas\$}p_1$

$a_4\texttt{nanas\$panama}b_1$

$a_5\texttt{nas\$panamaba}n_2$

$a_6\texttt{s\$panamabana}n_3$

$b_1\texttt{ananas\$panam}a_1$

$m_1\texttt{abananas\$pan}a_2$

$n_1\texttt{amabananas\$p}a_3$

$n_2\texttt{anas\$panamab}a_4$

$n_3\texttt{as\$panamaban}a_5$

$p_1\texttt{anamabananas}\$_1$

$s_1\texttt{\$panamabanan}a_6$

BWT Saves the Day Again

- searching for ana in panamabananas

$\$_1$panamabanana$s_1$

$a_1$bananas\$panam_1

$a_2$mabananas\$pan_1

$a_3$namabananas\$$p_1$

$a_4$nanas\$panamab_1

$a_5$nas\$panamaban_2

a_6s\$panamabanan_3

$b_1$ananas\$panama_1

$m_1$abananas\$pana_2

$n_1$amabananas\$pa_3

$n_2$anas\$panamaba_4

$n_3$as\$panamabana_5

$p_1$anamabananas$\$_1$

s_1\$panamabanana_6

BWT Saves the Day Again

- searching for a**na** in panamabananas

$\mathtt{\$_1panamabananas_1}$

$\mathtt{a_1bananas\$panam_1}$

$\mathtt{a_2mabananas\$pan_1}$

$\mathtt{a_3namabananas\$p_1}$

$\mathtt{a_4nanas\$panamab_1}$

$\mathtt{a_5nas\$panamaban_2}$

$\mathtt{a_6s\$panamabanan_3}$

$\mathtt{b_1ananas\$panama_1}$

$\mathtt{m_1abananas\$pana_2}$

$\mathtt{n_1amabananas\$pa_3}$

$\mathtt{n_2anas\$panamaba_4}$

$\mathtt{n_3as\$panamabana_5}$

$\mathtt{p_1anamabananas\$_1}$

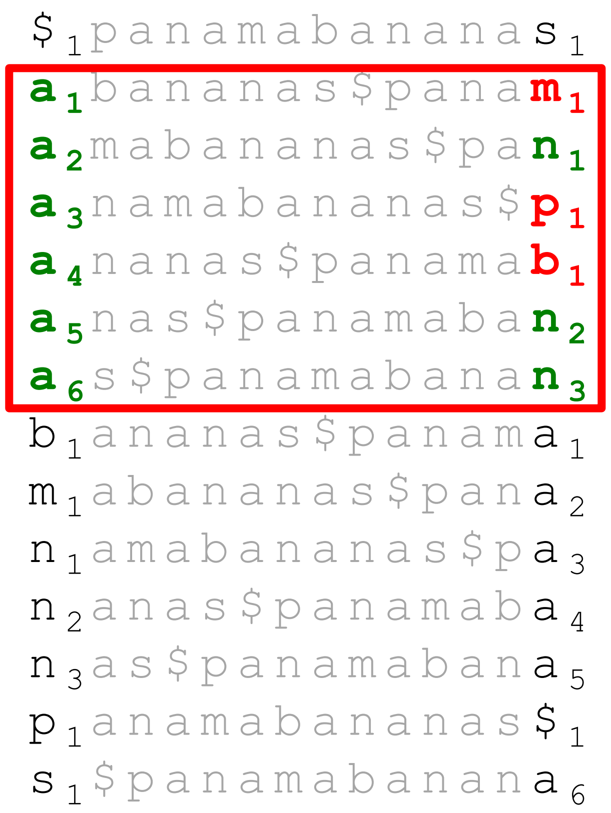
$\mathtt{s_1\$panamabanana_6}$

Exact matching

BWT Pattern Matching with 1 Mismatch

- searching for a**na** in panamabananas

To allow for 1 mismatch, we need to analyze the rows ending in red letters as well.



$\$_1\,\text{panamabanana}\,s_1$

$a_1\,\text{bananas\$pana}\,m_1$

$a_2\,\text{mabananas\$pa}\,n_1$

$a_3\,\text{namabananas\$}\,p_1$

$a_4\,\text{nanas\$panama}\,b_1$

$a_5\,\text{nas\$panamaba}\,n_2$

$a_6\,\text{s\$panamabana}\,n_3$

$b_1\,\text{ananas\$panam}\,a_1$

$m_1\,\text{abananas\$pan}\,a_2$

$n_1\,\text{amabananas\$p}\,a_3$

$n_2\,\text{anas\$panamab}\,a_4$

$n_3\,\text{as\$panamaban}\,a_5$

$p_1\,\text{anamabananas}\,\$_1$

$s_1\,\text{\$panamabanan}\,a_6$

Approximate matching with at most 1 mismatch

BWT Pattern Matching with 1 Mismatch

- searching for a**na** in panamabananas

To allow for 1 mismatch, we need to analyze the rows ending in red letters as well.

			# Mismatches
$\$_1$	panamabanana	s_1	
$\mathbf{a}_1$	bananas\$pana	$\mathbf{m}_1$	1
$\mathbf{a}_2$	mabananas\$pa	$\mathbf{n}_1$	0
$\mathbf{a}_3$	namabananas\$	$\mathbf{p}_1$	1
$\mathbf{a}_4$	nanas\$panama	$\mathbf{b}_1$	1
$\mathbf{a}_5$	nas\$panamaba	$\mathbf{n}_2$	0
$\mathbf{a}_6$	s\$panamabana	$\mathbf{n}_3$	0
b_1	ananas\$panam	a_1	
m_1	abananas\$pan	a_2	
n_1	amabananas\$p	a_3	
n_2	anas\$panamab	a_4	
n_3	as\$panamaban	a_5	
p_1	anamabananas	$\$_1$	
s_1	\$panamabanan	a_6	

BWT Pattern Matching with 1 Mismatch

- searching for a**na** in panamabananas

Now we analyze all rows with at most 1 mismatch using the First-Last property.

			# Mismatches
$\$_1$	panamabanana	s_1	
$\mathbf{a}_1$	bananas\$panam	$\mathbf{m}_1$	1
$\mathbf{a}_2$	mabananas\$pan	$\mathbf{n}_1$	0
$\mathbf{a}_3$	namabananas\$	$\mathbf{p}_1$	1
$\mathbf{a}_4$	nanas\$panama	$\mathbf{b}_1$	1
$\mathbf{a}_5$	nas\$panamaban	$\mathbf{n}_2$	0
$\mathbf{a}_6$	s\$panamabanan	$\mathbf{n}_3$	0
b_1	ananas\$panam	a_1	
m_1	abananas\$pan	a_2	
n_1	amabananas\$p	a_3	
n_2	anas\$panamab	a_4	
n_3	as\$panamaban	a_5	
p_1	anamabananas	$\$_1$	
s_1	\$panamabanan	a_6	

BWT Pattern Matching with 1 Mismatch

- searching for ana in panamabananas

Now we analyze all rows with at most 1 mismatch using the First-Last property.

	# Mismatches
$\$_1$panamabanana$s_1$	
$a_1$bananas\$panam_1	1
$a_2$mabananas\$pan_1	0
$a_3$namabananas\$$p_1$	1
$a_4$nanas\$panamab_1	1
$a_5$nas\$panamaban_2	0
a_6s\$panamabanan_3	0
$b_1$ananas\$panama_1	
$m_1$abananas\$pana_2	
$n_1$amabananas\$pa_3	
$n_2$anas\$panamaba_4	
$n_3$as\$panamabana_5	
$p_1$anamabananas$\$_1$	
s_1\$panamabanana_6	

BWT Pattern Matching with 1 Mismatch

- searching for a**na** in panamabananas

Now we analyze all rows with at most 1 mismatch using the First-Last property.

			# Mismatches
$\$_1$	panamabanana	s_1	
a_1	bananas$pana	m_1	
a_2	mabananas$pa	n_1	
a_3	namabananas$	p_1	
a_4	nanas$panama	b_1	
a_5	nas$panamaba	n_2	
a_6	s$panamabana	n_3	
b_1	ananas$panam	a_1	1
m_1	abananas$pan	a_2	1
n_1	amabananas$p	a_3	0
n_2	anas$panamab	a_4	0
n_3	as$panamaban	a_5	0
p_1	anamabananas	$\$_1$	1
s_1	$panamabanan	a_6	

BWT Pattern Matching with 1 Mismatch

- searching for **ana** in panamabananas

			# Mismatches
$\$_1$	panamabanana	s_1	
a_1	bananas\$pana	m_1	
a_2	mabananas\$pa	n_1	
a_3	namabananas\$	p_1	
a_4	nanas\$panama	b_1	
a_5	nas\$panamaba	n_2	
a_6	s\$panamabana	n_3	
$\mathbf{b_1}$ **a**	nanas\$panam	**a_1**	1
$\mathbf{m_1}$ **a**	bananas\$pan	**a_2**	1
$\mathbf{n_1}$ **a**	mabananas\$p	**a_3**	0
$\mathbf{n_2}$ **a**	nas\$panamab	**a_4**	0
$\mathbf{n_3}$ **a**	s\$panamaban	**a_5**	0
$\mathbf{p_1}$ **a**	namabananas	**$\$_1$**	**2**
s_1	\$panamabanan	a_6	

This row results in a 2nd mismatch (the $), so we discard it.

Five Approximate Matches Found!

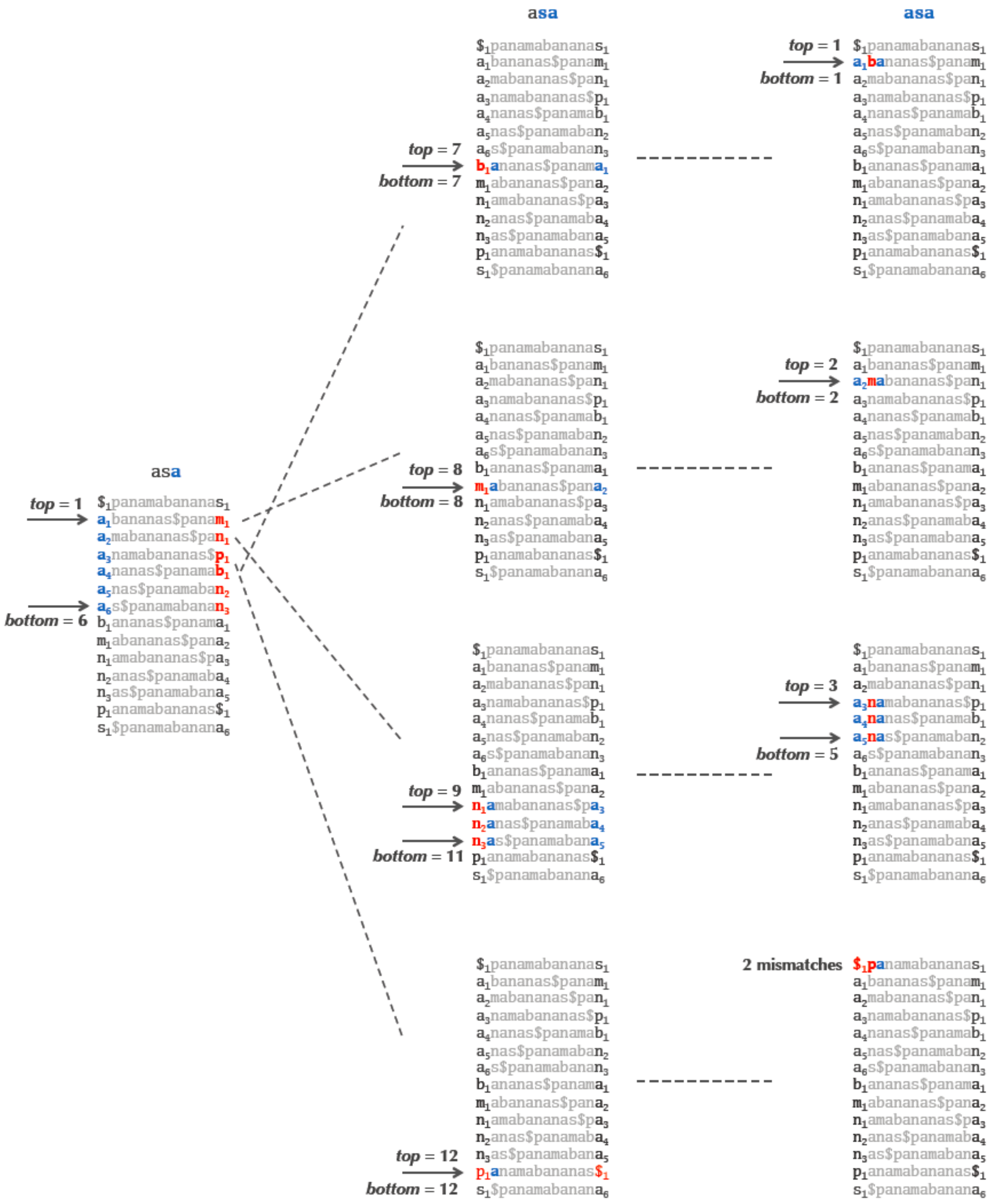
- searching for **ana** in panamabananas

			# Mismatches
$\$_1$	panamabanana	s_1	
a_1	**b**ananas$pana	m_1	1
a_2	**m**abananas$pa	n_1	1
a_3	**n**amabananas$	p_1	0
a_4	**n**anas$panama	b_1	0
a_5	**n**as$panamaba	n_2	0
a_6	s$panamabana	n_3	
b_1	ananas$panam	a_1	
m_1	abananas$pan	a_2	
n_1	amabananas$p	a_3	
n_2	anas$panamab	a_4	
n_3	as$panamaban	a_5	
p_1	anamabananas	$\$_1$	
s_1	$panamabanan	a_6	

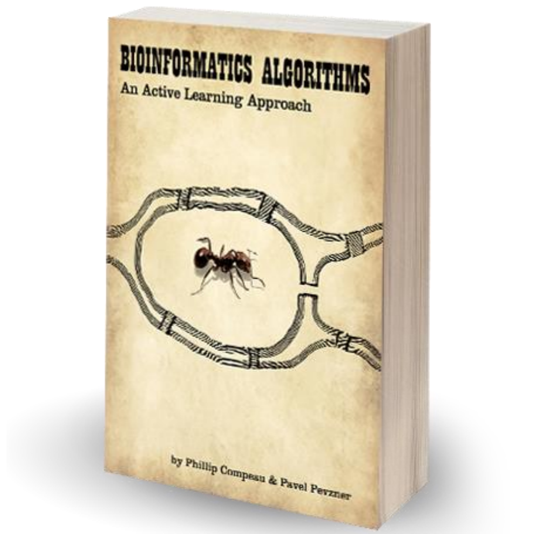
Where Are The Matches?

- searching for **ana** in panamabananas

	Suffix Array
$\$_1$panamabanana$s_1$	
$a_1$bananas\$panam_1	5
$a_2$mabananas\$pan_1	3
$a_3$namabananas\$$p_1$	1
$a_4$nanas\$panamab_1	7
$a_5$nas\$panamaban_2	9
a_6s\$panamabanan_3	
$b_1$ananas\$panama_1	
$m_1$abananas\$pana_2	
$n_1$amabananas\$pa_3	
$n_2$anas\$panamaba_4	
$n_3$as\$panamabana_5	
$p_1$anamabananas$\$_1$	
s_1\$panamabanana_6	



In reality, approximate pattern matching with BWT is more complex (we omitted various details)


BIOINFORMATICS ALGORITHMS
An Active Learning Approach
by Phillip Compeau & Pavel Pevzner