
Burrows-Wheeler Transformation

1. Append to the input string a special char, \$, smaller than all alphabet.

mississippi\$

Burrows-Wheeler Transformation (cnt'd)

2. Generate all rotations.

m	i	s	s	i	s	s	i	p	p	i	\$
i	s	s	i	s	s	i	p	p	i	\$	m
s	s	i	s	s	i	p	p	i	\$	m	i
s	i	s	s	i	p	p	i	\$	m	i	s
i	s	s	i	p	p	i	\$	m	i	s	s
s	s	i	p	p	i	\$	m	i	s	s	i
s	i	p	p	i	\$	m	i	s	s	i	s
i	p	p	i	\$	m	i	s	s	i	s	s
p	p	i	\$	m	i	s	s	i	s	s	i
p	i	\$	m	i	s	s	i	s	s	i	p
i	\$	m	i	s	s	i	s	s	i	p	p
\$	m	i	s	s	i	s	s	i	p	p	i

Burrows-Wheeler Transformation (cnt'd)

- Sort rotations according to the alphabetical order.

\$	m	i	s	s	i	s	s	i	p	p	i
i	\$	m	i	s	s	i	s	s	i	p	p
i	p	p	i	\$	m	i	s	s	i	s	s
i	s	s	i	p	p	i	\$	m	i	s	s
i	s	s	i	s	s	i	p	p	i	\$	m
m	i	s	s	i	s	s	i	p	p	i	\$
p	i	\$	m	i	s	s	i	s	s	i	p
p	p	i	\$	m	i	s	s	i	s	s	i
s	i	p	p	i	\$	m	i	s	s	i	s
s	i	s	s	i	p	p	i	\$	m	i	s
s	s	i	p	p	i	\$	m	i	s	s	i
s	s	i	s	s	i	p	p	i	\$	m	i

Burrows-Wheeler Transformation (cnt'd)

4. Output
the last
column.

\$	m	i	s	s	i	s	s	i	p	p	i
i	\$	m	i	s	s	i	s	s	i	p	p
i	p	p	i	\$	m	i	s	s	i	s	s
i	s	s	i	p	p	i	\$	m	i	s	s
i	s	s	i	s	s	i	p	p	i	\$	m
m	i	s	s	i	s	s	i	p	p	i	\$
p	i	\$	m	i	s	s	i	s	s	i	p
p	p	i	\$	m	i	s	s	i	s	s	i
s	i	p	p	i	\$	m	i	s	s	i	s
s	i	s	s	i	p	p	i	\$	m	i	s
s	s	i	p	p	i	\$	m	i	s	s	i
s	s	i	s	s	i	p	p	i	\$	m	i

Burrows-Wheeler Transformation (cnt'd)

mississippi\$



ipssm\$piissii

BWT - alternative construction

T = a b a a b a

BWT

\$ a b a a b a
a \$ a b a a b
a a b a \$ a b
a b a \$ a b a
a b a a b a \$
b a \$ a b a a
b a a b a \$ a

Suffix Array

6 \$
5 a \$
2 a a b a \$
3 a b a \$
0 a b a a b a \$
4 b a \$
1 b a a b a \$

$$\text{BWT}[i] = \begin{cases} T[\text{SA}[i] - 1], & \text{if SA}[i] > 0 \\ \$, & \text{if SA}[i] = 0 \end{cases}$$

BWT = characters just to the left of characters in SA

Ferragina-Manzini Index

First column: F

Last column: L

Let's make an
L to F map.

Observation:
The n^{th} i in L is
the n^{th} i in F.

\$	m	i	s	s	i	s	s	i	p	p	i
i	\$	m	i	s	s	i	s	s	i	p	p
i	p	p	i	\$	m	i	s	s	i	s	s
i	s	s	i	p	p	i	\$	m	i	s	s
i	s	s	i	s	s	i	p	p	i	\$	m
m	i	s	s	i	s	s	i	p	p	i	\$
p	i	\$	m	i	s	s	i	s	s	i	p
p	p	i	\$	m	i	s	s	i	s	s	i
s	i	p	p	i	\$	m	i	s	s	i	s
s	i	s	s	i	p	p	i	\$	m	i	s
s	s	i	p	p	i	\$	m	i	s	s	i
s	s	i	s	s	i	p	p	i	\$	m	i

Ferragina-Manzini Index: L to F map

Store/compute a two dimensional $\text{Occ}(j, 'c')$ table of the number of occurrences of char 'c' up to position j (inclusive).

and one dimensional $\text{Cnt}('c')$ and $\text{Rank}('c')$ tables

	\$	i	m	p	s
i	0	1	0	0	0
p	0	1	0	1	0
s	0	1	0	1	1
s	0	1	0	1	2
m	0	1	1	1	2
\$	1	1	1	1	2
p	1	1	1	2	2
i	1	2	1	2	2
s	1	2	1	2	3
s	1	2	1	2	4
i	1	3	1	2	4
i	1	4	1	2	4

$\text{Occ}(j, 'c')$

$\text{Cnt}('c')$

\$	i	m	p	s
1	4	1	2	4

$\text{Rank}('c')$

\$	i	m	p	s
12	2	1	9	3

Ferragina-Manzini Index: L to F map

$$\begin{aligned}
 & [\text{Cnt}(\$) + \\
 & \text{Cnt}(i) + \\
 & \text{Cnt}(m) + \\
 & \text{Cnt}(p) = 8] \\
 & + [\text{Occ}(9, 's') = 3] \\
 & = 11
 \end{aligned}$$

1	\$	m	i	s	s	i	s	s	i	p	p	i
2	i	\$	m	i	s	s	i	s	s	i	p	p
3	i	p	p	i	\$	m	i	s	s	i	s	s
4	i	s	s	i	p	p	i	\$	m	i	s	s
5	i	s	s	i	s	s	i	p	p	i	\$	m
6	m	i	s	s	i	s	s	i	p	p	i	\$
7	p	i	\$	m	i	s	s	i	s	s	i	p
8	p	p	i	\$	m	i	s	s	i	s	s	i
9	s	i	p	p	i	\$	m	i	s	s	i	s
10	s	i	s	s	i	p	p	i	\$	m	i	s
11	s	s	i	p	p	i	\$	m	i	s	s	i
12	s	s	i	s	s	i	p	p	i	\$	m	i

before 's' →

's' section →

Cnt('c')

\$	i	m	p	s
1	4	1	2	4

Ferragina-Manzini Index: Reverse traversal

- (1) i
- (2) p
- (7) p
- (8) i
- (3) s
- (9) s
- (11) i
- (4) s
- (10) s
- (12) i
- (5) m
- (6) \$

1	\$	m	i	s	s	i	s	s	i	p	p	i
2	i	\$	m	i	s	s	i	s	s	i	p	p
3	i	p	p	i	\$	m	i	s	s	i	s	s
4	i	s	s	i	p	p	i	\$	m	i	s	s
5	i	s	s	i	s	s	i	p	p	i	\$	m
6	m	i	s	s	i	s	s	i	p	p	i	\$
7	p	i	\$	m	i	s	s	i	s	s	i	p
8	p	p	i	\$	m	i	s	s	i	s	s	i
9	s	i	p	p	i	\$	m	i	s	s	i	s
10	s	i	s	s	i	p	p	i	\$	m	i	s
11	s	s	i	p	p	i	\$	m	i	s	s	i
12	s	s	i	s	s	i	p	p	i	\$	m	i

Search with BWT-FM

Auxillary data structures for efficient pattern matching:
how to find the corresponding chars in the first column
efficiently, in terms of both time and space.

	a	c	g	t
rank	1	5	8	11

Original sequence

BWT

	SA	Original sequence	BWT	a	c	g	t
		\$agcagcagact	t	0	0	0	1
1	9	act\$agcagcag	g	0	0	1	1
2	7	agact\$agcagc	c	0	1	1	1
3	4	agcagact\$agc	c	0	2	1	1
4	1	agcagcagact\$	\$	0	2	1	1
5	6	cagact\$agcag	g	0	2	2	1
6	3	cagcagact\$ag	g	0	2	3	1
7	10	ct\$agcagcaga	a	1	2	3	1
8	8	gact\$agcagca	a	2	2	3	1
9	5	gcagact\$agca	a	3	2	3	1
10	2	gcagcagact\$a	a	4	2	3	1
11	11	t\$agcagcagac	c	4	3	3	1

FM indices

Search with BWT-FM

Auxillary data structures for efficient pattern matching: how to find the corresponding chars in the first column efficiently, in terms of both time and space.

	a	c	g	t
rank	1	5	8	11

Original sequence

BWT

	SA	Original sequence	BWT	a	c	g	t
		gca					
		\$agcagcagact	t	0	0	0	1
1	9	act\$agcagcag	g	0	0	1	1
2	7	agact\$agcagc	c	0	1	1	1
3	4	agcagact\$agc	c	0	2	1	1
4	1	agcagcagact\$	\$	0	2	1	1
5	6	cagact\$agcag	g	0	2	2	1
6	3	cagcagact\$ag	g	0	2	3	1
7	10	ct\$agcagcaga	a	1	2	3	1
8	8	gact\$agcagca	a	2	2	3	1
9	5	gcagact\$agca	a	3	2	3	1
10	2	gcagcagact\$a	a	4	2	3	1
11	11	t\$agcagcagac	c	4	3	3	1

FM indices

Next block:
From $1 + 0 = 1$
to $1 + (4-1) = 4$

Search with BWT-FM

Auxillary data structures for efficient pattern matching: how to find the corresponding chars in the first column efficiently, in terms of both time and space.

	a	c	g	T
rank	1	5	8	11

Original sequence

BWT

	SA	Original sequence	BWT	a	c	g	t
		gca					
		\$agcagcagact	t	0	0	0	1
1	9	act\$agcagcag	g	0	0	1	1
2	7	agact\$agcagc	c	0	1	1	1
3	4	agcagact\$agc	c	0	2	1	1
4	1	agcagcagact\$	\$	0	2	1	1
5	6	cagact\$agcag	g	0	2	2	1
6	3	cagcagact\$ag	g	0	2	3	1
7	10	ct\$agcagcaga	a	1	2	3	1
8	8	gact\$agcagca	a	2	2	3	1
9	5	gcagact\$agca	a	3	2	3	1
10	2	gcagcagact\$a	a	4	2	3	1
11	11	t\$agcagcagac	c	4	3	3	1

FM indices

Next block:
From $5 + 0 = 5$
to $5 + (2-1) = 6$

Search with BWT-FM

Auxillary data structures for efficient pattern matching: how to find the corresponding chars in the first column efficiently, in terms of both time and space.

	a	c	g	T
rank	1	5	8	11

Original sequence

BWT

	SA	Original sequence	BWT	a	c	g	t
		gca	t	0	0	0	1
1	9	act\$agcagcag	g	0	0	1	1
2	7	agact\$agcagc	c	0	1	1	1
3	4	agcagact\$agc	c	0	2	1	1
4	1	agcagcagact\$	\$	0	2	1	1
5	6	cagact\$agcag	g	0	2	2	1
6	3	cagcagact\$ag	g	0	2	3	1
7	10	ct\$agcagcaga	a	1	2	3	1
8	8	gact\$agcagca	a	2	2	3	1
9	5	gcagact\$agca	a	3	2	3	1
10	2	gcagcagact\$a	a	4	2	3	1
11	11	t\$agcagcagac	c	4	3	3	1

FM indices

Next block:
From $8 + (2-1) = 9$
to $8 + (3-1) = 10$

Inexact match

