Burrows-Wheeler Transform

Wisely

Introduction of BWT

- Burrows and Wheeler introduced a new compression algorithm based on a reversible transformation now called the *Burrows-Wheeler Transform* (BWT)

- BWT is applied in data compression techniques such as bzip2 (http://bzip.org/)

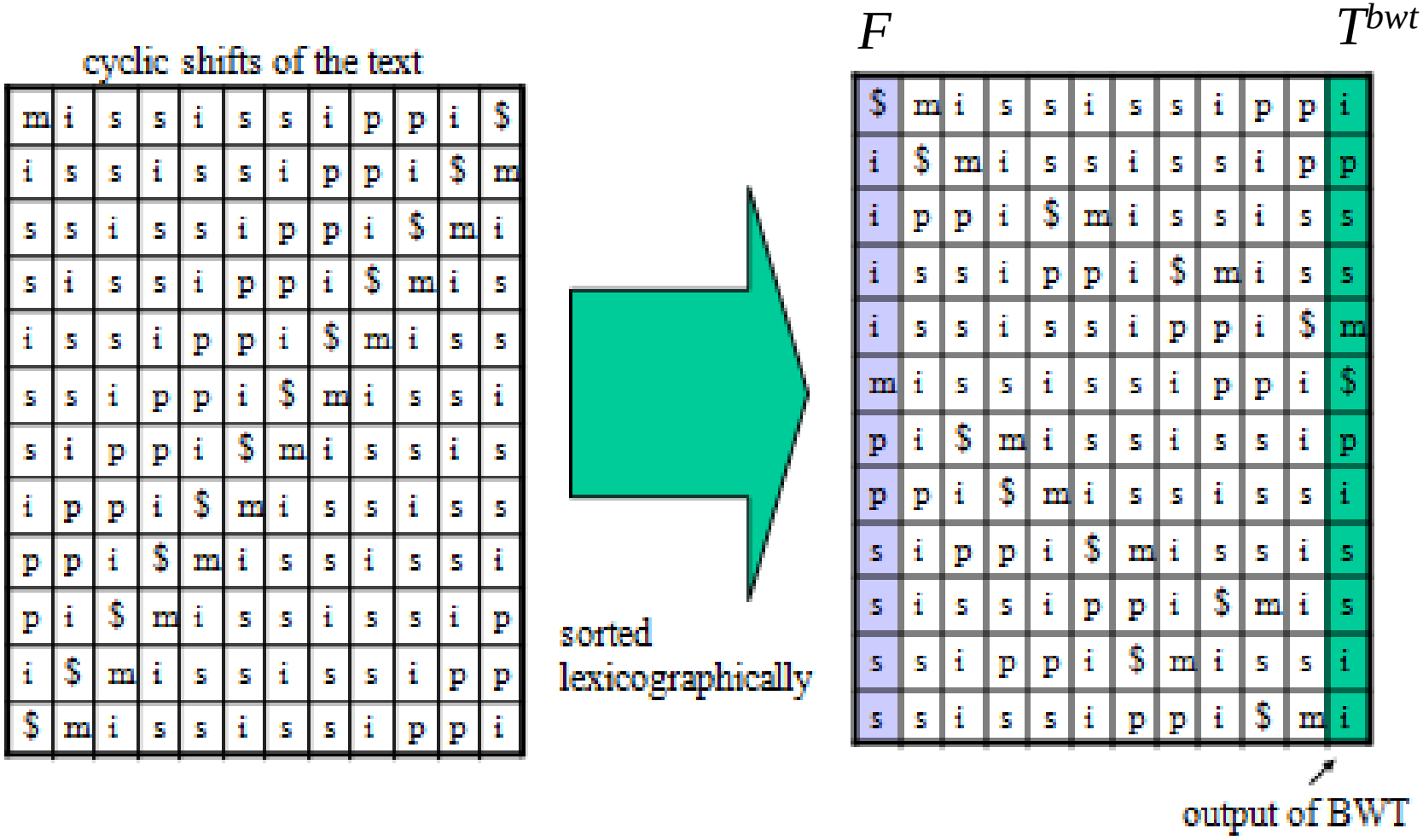
Transform Steps

(1) Append at the end of a text T a special character \$ smaller than any other text character

(2) Form a conceptual matrix M_T whose rows are the cyclic shifts of the string T\$ sorted in lexicographic order

(3) Construct the transformed text T^{bwt} by taking the last column of matrix M_T

Example :



BWT sorts the characters by their context

Notation

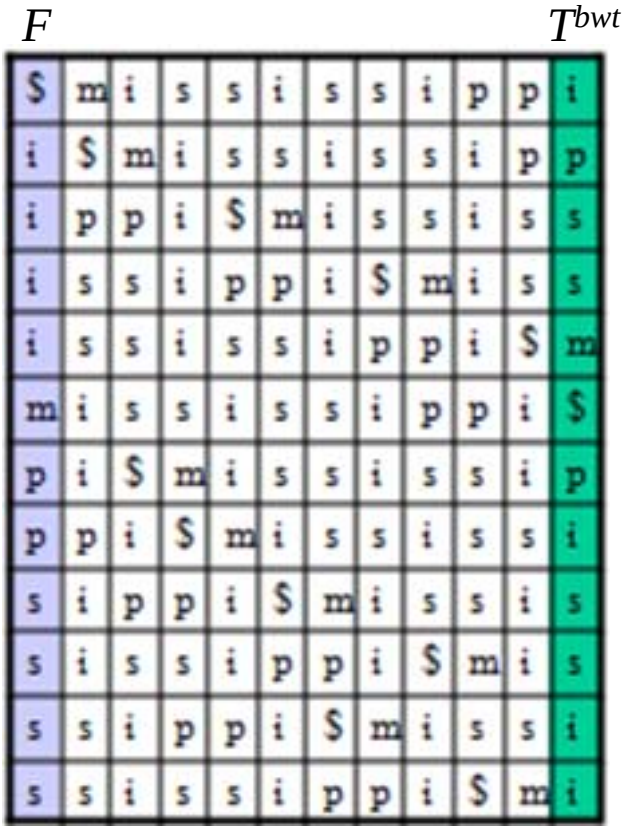
(1) Let $C[\ ]$ be an array of length $|\Sigma|$ such that

$C[c]$ = total # of text characters which are alphabetically smaller than c

(2) Let Occ(c,q) denote # of occurrences of character c in the prefix $T^{bwt}[1, q]$.

(3) Let $LF(i) = C[\ T^{bwt}[i]\ ] + \text{Occ}(T^{bwt}[i]\ ,\ i\ )$

Example :



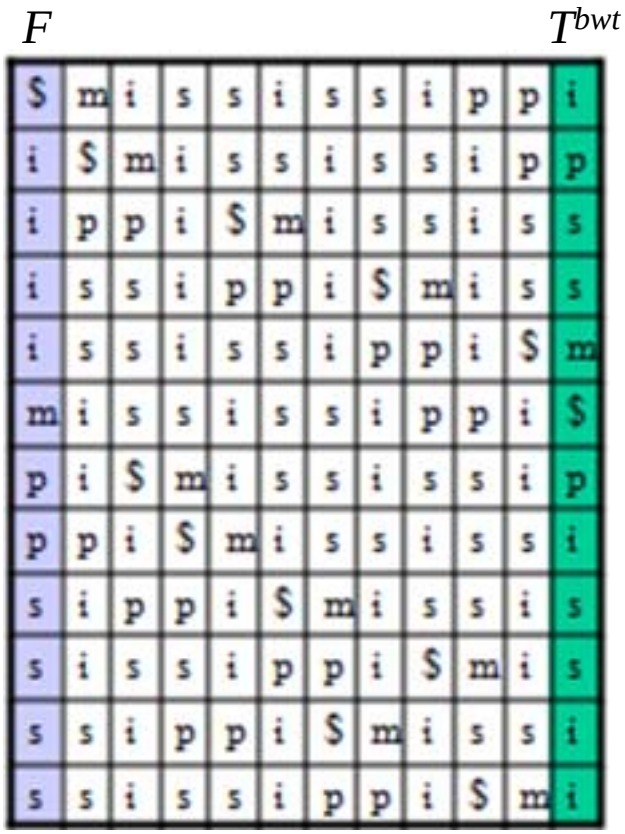
C table

$\$$	i	m	p	s
0	1	5	6	8

Occ(c,q)

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

Last to Front Mapping



- *LF*( ) = Last-to-Front Column Mapping
  - The character $T^{bwt}[i]$ is located in the first column *F* at position *LF*[*i*]
- *LF*(10) = *C*[*s*] + Occ(*s*,10)=12.
  - Both $T^{bwt}[10]$ and *F*[12] correspond to the first *s* in the *mississippi*

Backward Search

- The *LF*( ) mapping allows us to scan the text *T* backward.
- In other words, we could search a pattern in *T* backward. (How?)

Backward Search Algorithm

```
Backward_Search( P[1,p]  )
{
  i = p, c = P[p], First = C[c]+1, Last = C[c+1];
  while ( ( First ≦ Last) and i ≧ 2 ) {
     c = P[i-1];
     First = C[c] + Occ(c, First-1)+1;
     Last  = C[c] + Occ(c, Last);
     i = i-1;
  }
  if ( Last < First ) then
       return “no occurrence” ;
  else return ( First, Last );
}
```

References

[1] M. Burrows and D. Wheeler (1994).

A Block Sorting Lossless Data Compression Algorithm.

Technical Report 124, Digital Equipment Corporation.

[2] G. Manzini (2001).

An Analysis of the Burrows-Wheeler Transform.

Journal of the ACM, 48(3): pages 407 – 430.

[3] P. Ferragina and G. Manzini (2000).

Opportunistic Data Structures with Applications.

In Proceedings of FOCS, pages 390 – 398.

[4] P. Ferragina, G. Manzini, V. Mäkinen, and G. Navarro (2004).

An Alphabet-Friendly FM-Index.

In Proceedings of SPIRE, pages 150 – 160.