Web Data Compression and Search

Fast BWT Construction

Space Efficient Linear Time Construction of Suffix Arrays

A good paper by Pang Ko and Srinivas Aluru

J. Discrete Algorithms 3(2-4): 143-156(2005)

Suffix Array

- Sorted order of suffixes of a string *T*.
- Represented by the starting position of the suffix.

Text	M	I	S	S	I	S	S	I	P	P	I	$
Index	1	2	3	4	5	6	7	8	9	10	11	12
Suffix Array	12	11	8	5	2	1	10	9	7	4	6	3

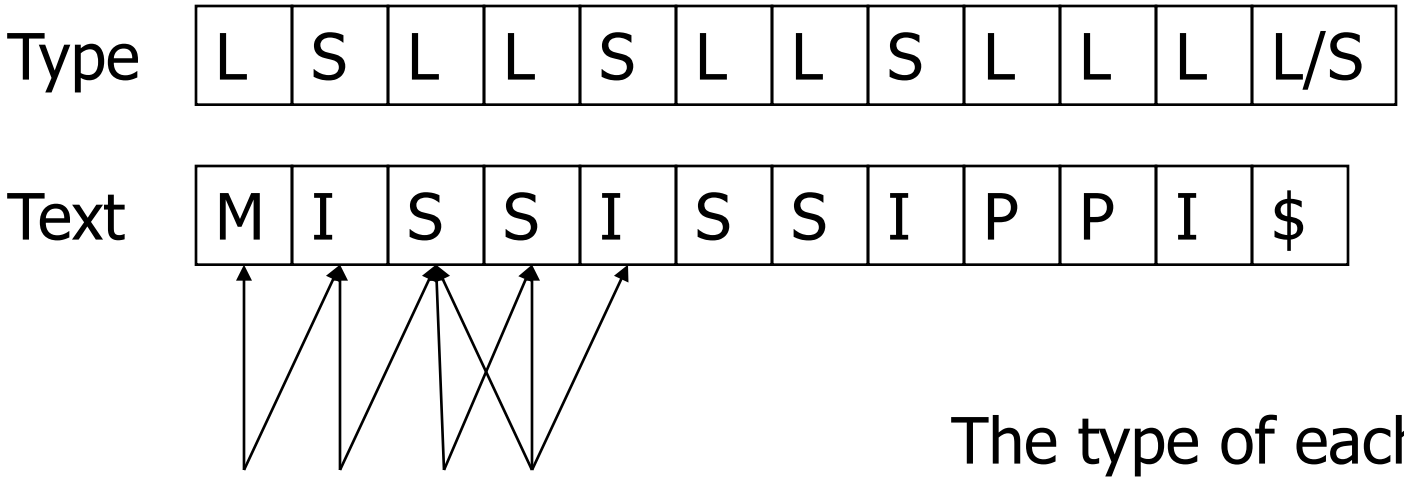
Notation

- String $T = t_1 \ldots t_n$.
- Over the alphabet $\Sigma = \{1 \ldots n\}$.
- $t_n =$ '\$' , '\$' is a unique character.
- $T_i = t_i \ldots t_n$, denotes the i-th suffix of T.
- For strings α and β, $\alpha < \beta$ denotes α is lexicographically smaller than β.

Overview

- Divide all suffixes of T into two types.
  - Type S suffixes = $\{T_i \mid T_i < T_{i+1}\}$
  - Type L suffixes = $\{T_j \mid T_j > T_{j+1}\}$
  - The last suffix is both type S and L.
- Sort all suffixes of one of the types.
- Obtain lexicographical order of all suffixes from the sorted ones.

Identify Suffix Types



M > I < S = S > I,

$T_1 > T_2$ so check next character.

T_1 is type L.

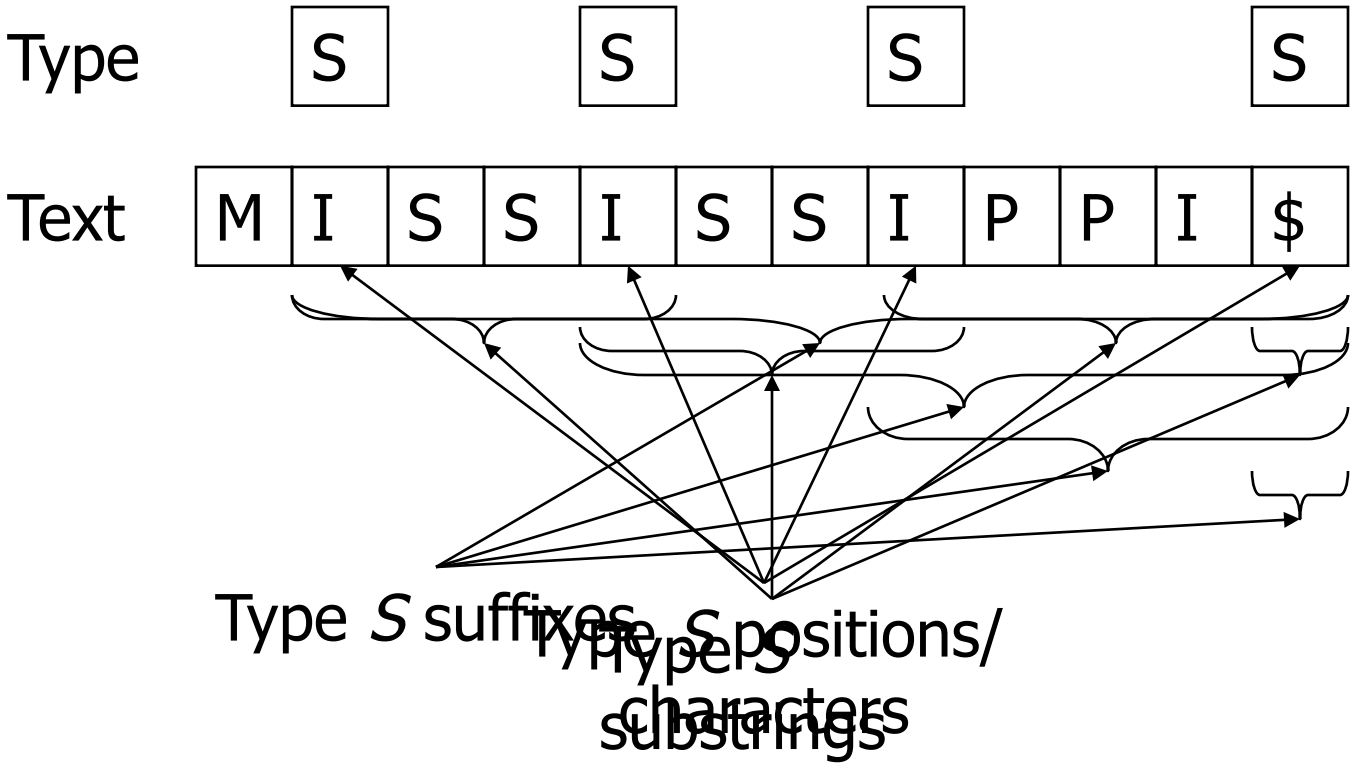
$T_2 < T_3$ so T_2 is type S.

$T_3 = T_4$ so check next character.

T_3 and T_4 are type L

The type of each suffix in T can be determined in one scan of the string.

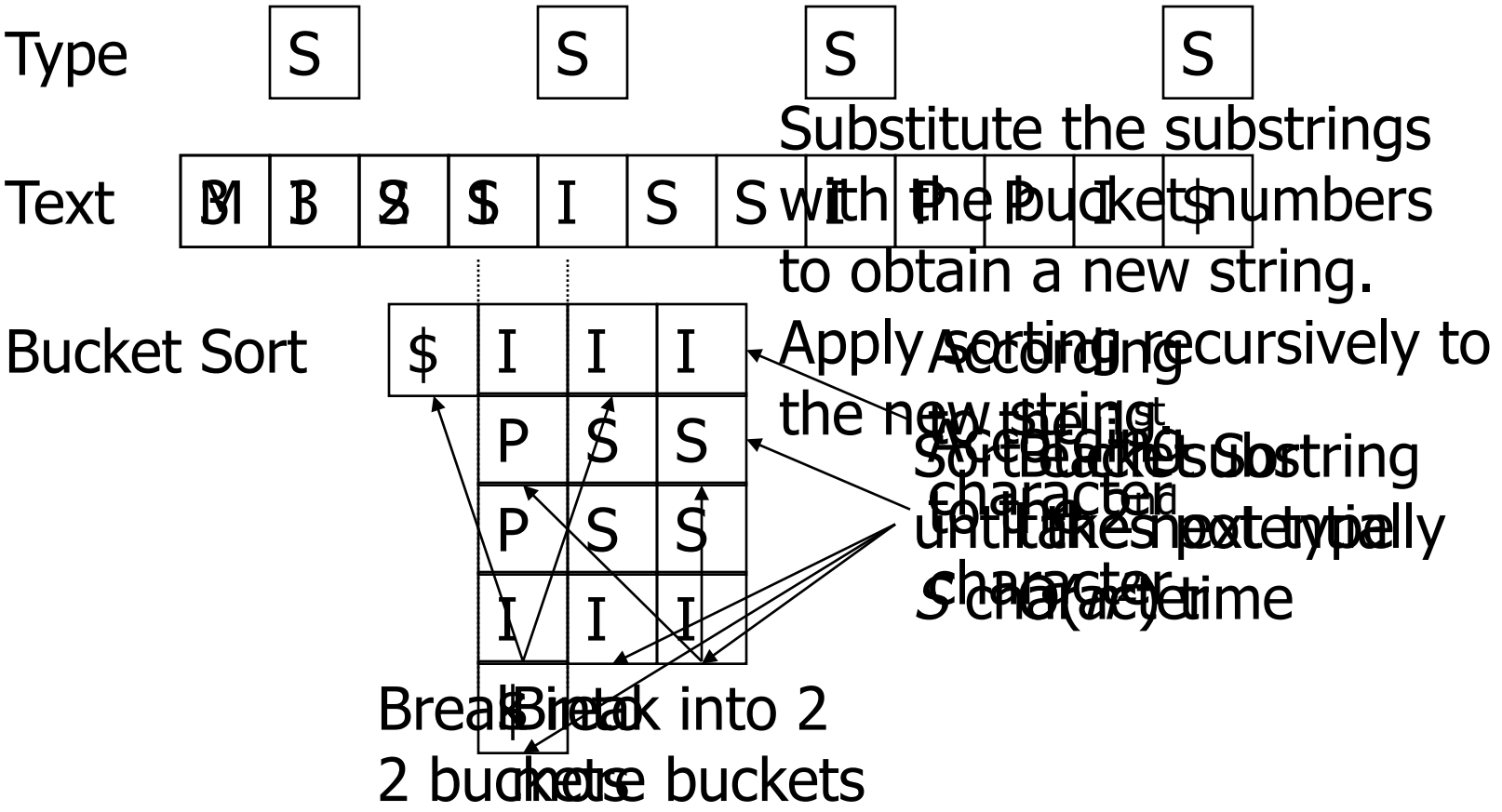
Notation


Type
S
S
S
S
Text
M
I
S
S
I
S
S
I
P
P
I
$
Type S suffixes
Type S positions/ characters
Type S substrings

Sorting Type *S* Suffixes

- Sort all type *S* substrings.
- Replace each type *S* substrings by its bucket number.
- New string is the sequence of bucket numbers.
- Sorting all type *S* suffixes = Sorting all suffixes of the new string.

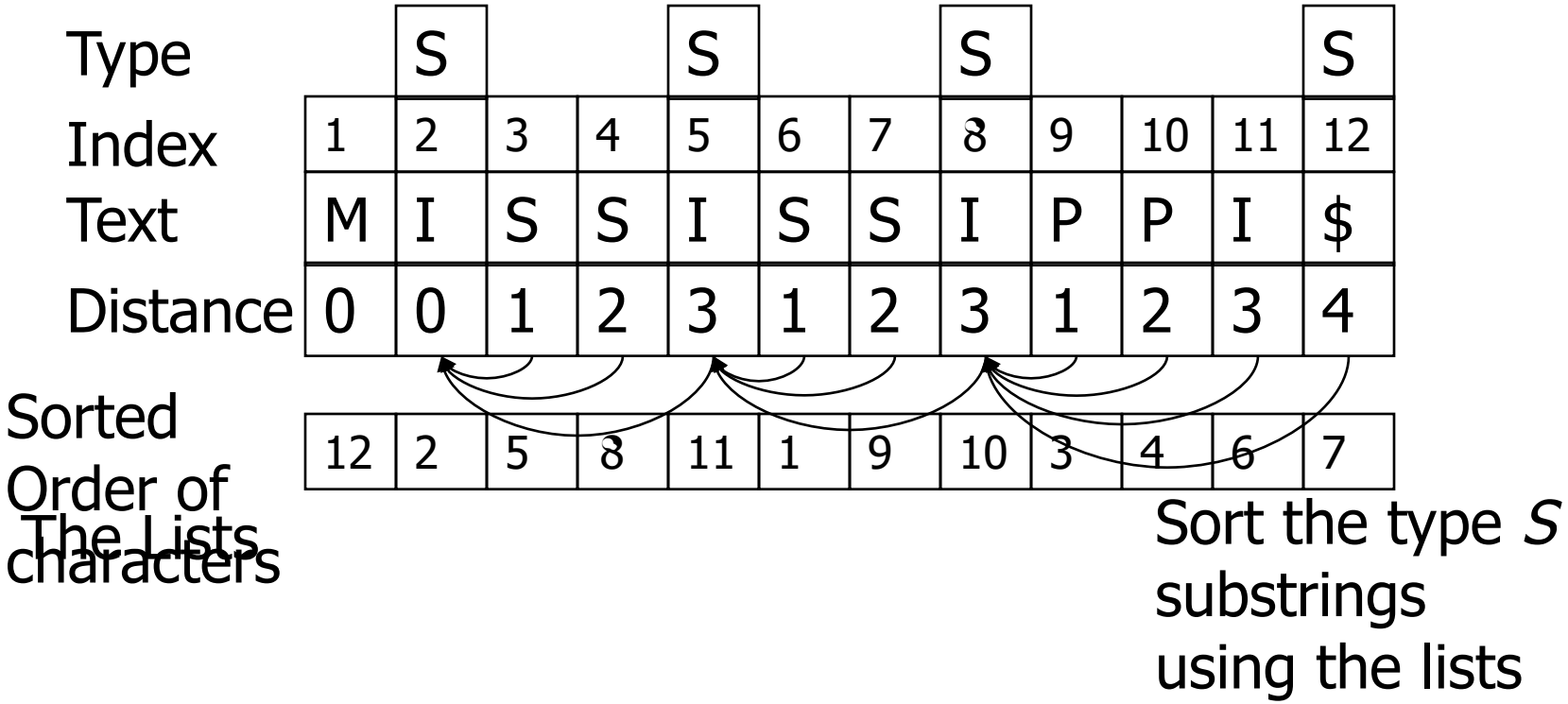
Sorting Type *S* Substrings



Solution

- Observation: Each character participates in the bucket sort at most twice.
  - Type *L* characters only participate in the bucket sort once.
- Solution:
  - Sort all the characters once.
  - Construct m lists according the distance to the closest type *S* character to the left

Illustration



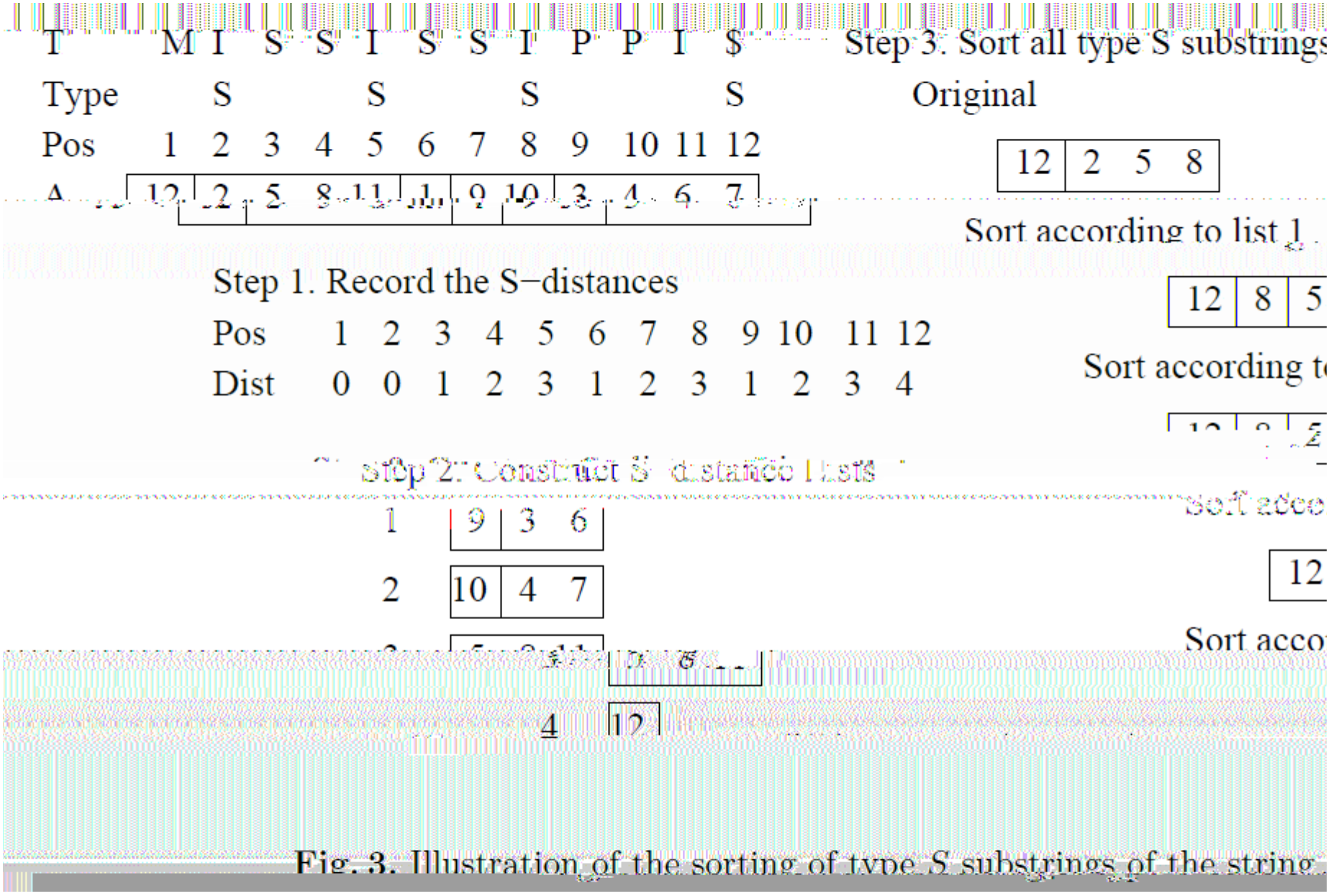
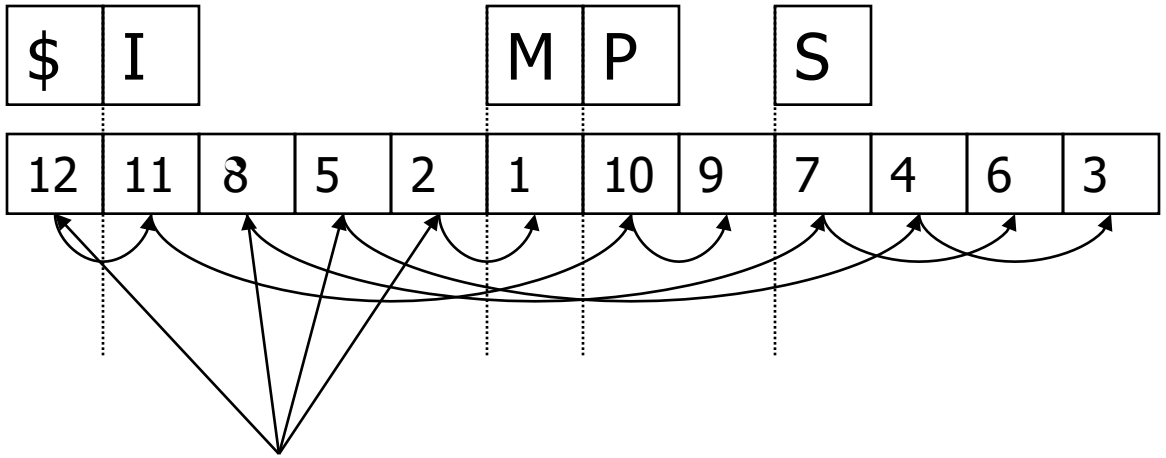


Fig. 3. Illustration of the sorting of type S substrings of the string

Construct Suffix Array for all Suffixes

- The first suffix in the suffix array is a type S suffix.
- For $1 \leq i \leq n$, if $T_{SA[i]-1}$ is type *L*, move it to the current front of its bucket
- [$:12][I:2,5,8,11][M:1][P:9,10][S:3,4,6,7]



Sorted order of type *S* suffixes

Exercise

- Consider the popular example string S:
- **$**

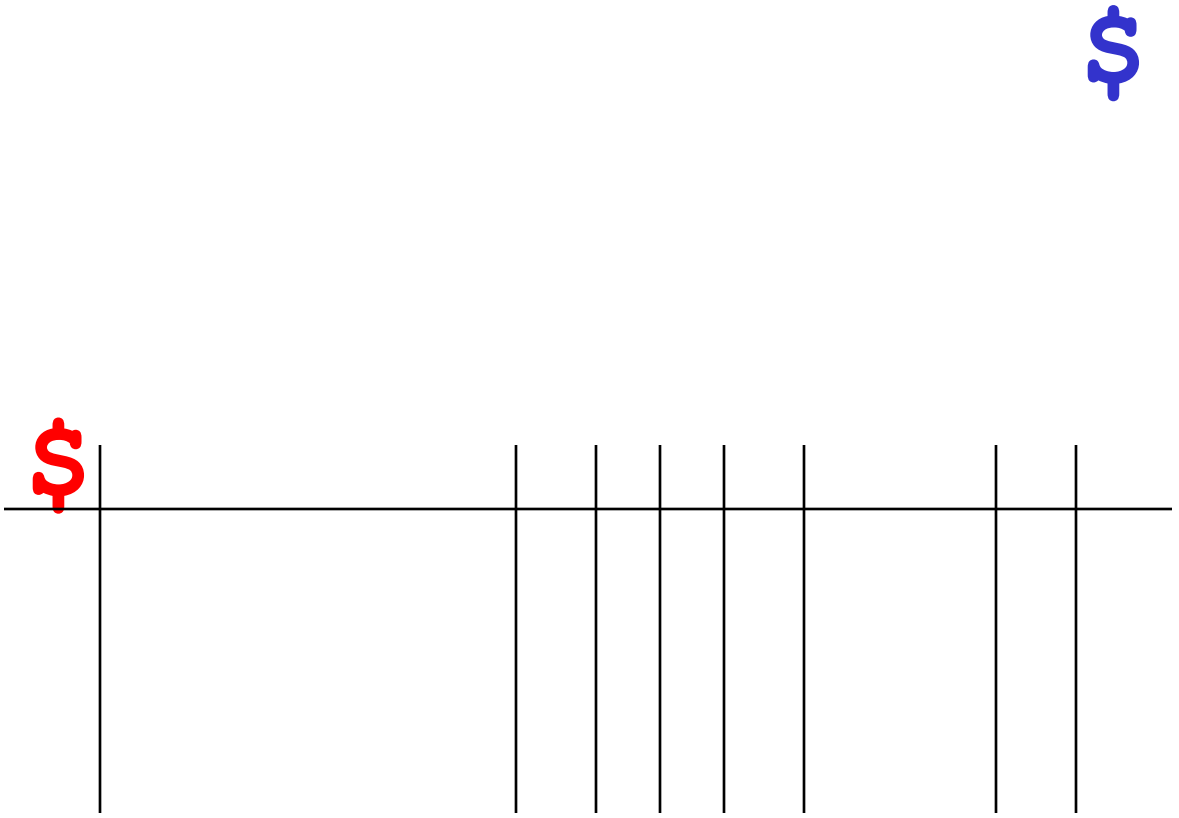
1. Construct the suffix array of S using the linear time algorithm
2. Then compute the BWT(S)
3. What's the relationship between the suffix array and BWT ?

Step – Identify the type of each suffix

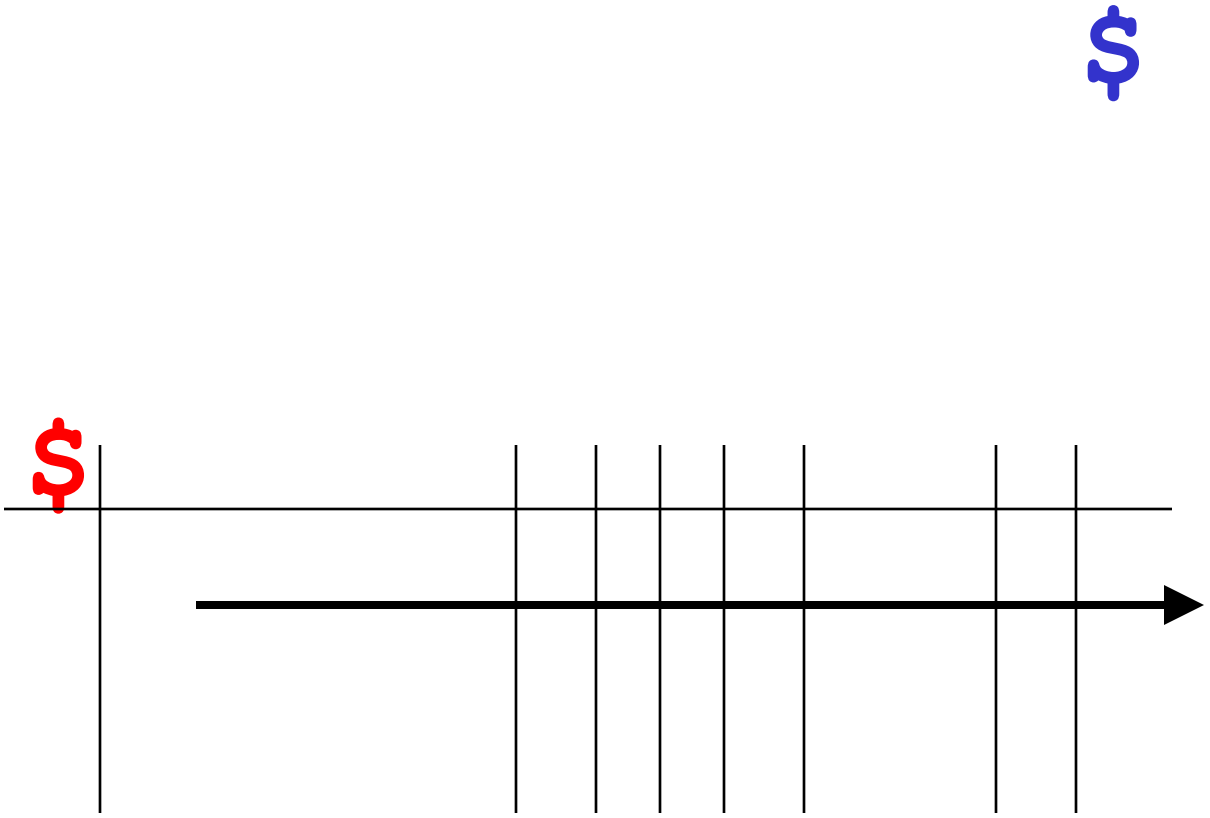
-
- $
-
-

Step – Compute the distance from S

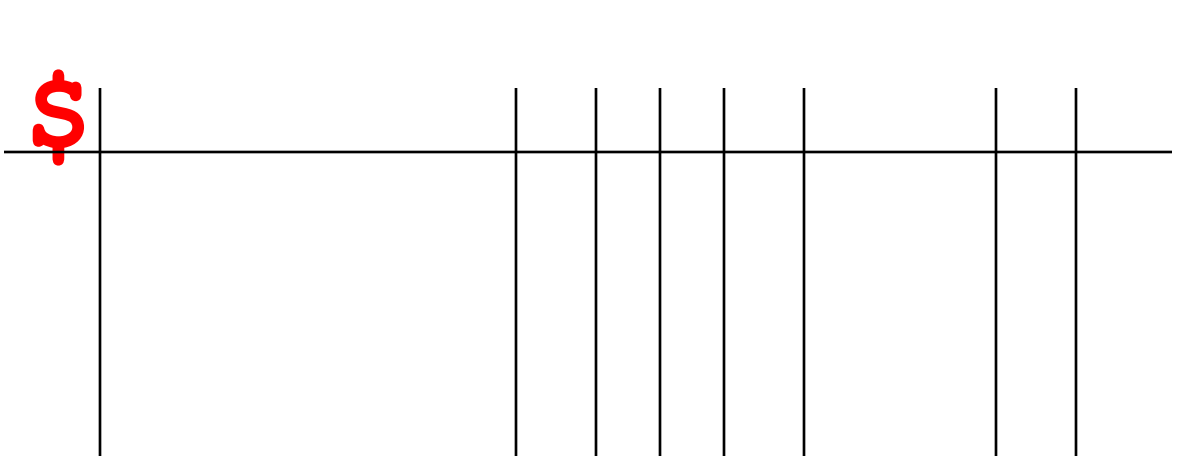
Step – Sort order of chars



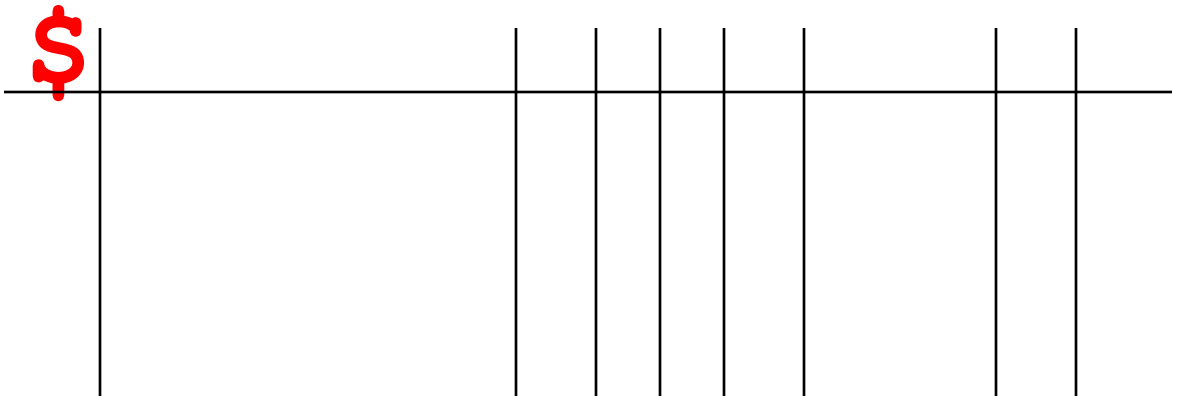
Step – Construct m-Lists



Step – Generate m-Lists



Step – Sort S substrings



Step – Sort S substrings

<footer>23</footer>

Step – Sort S substrings

Step – Generate the Suffix Array

Step – Generate the Suffix Array

- $
-
-
- $
-
-

Step – Generate the Suffix Array

- $
-
-
- $
-
-

Step – Generate the Suffix Array

- $
-
-
-
- $
-
-

Step – Generate the Suffix Array

- $
-
-
-
- $
-
-

Step – Generate the Suffix Array

- $
-
-
- $
-
-

Step – Generate the Suffix Array

- $
-
-
- $
-
-

Final answer

-
-
-
-
-
-

$

Final answer

-
-
-
-
-
-

$

What is the BWT(S) ?

BWT is easy!

-
-
-
-
-
-
-
-
-

$

BWT construction in linear time

- $
- $