9I   Construct the Burrows-Wheeler Transform of a String

Burrows-Wheeler Transform Construction Problem

Construct the Burrows-Wheeler transform of a string.

Input: A string *Text*.
Output: BWT(*Text*).

$BANANAS
ANANAS$B
ANAS$BAN
AS$BANAN
BANANAS$
NANAS$BA
NAS$BANA
S$BANANA

Formatting

Input: A string *Text*.
Output: A string *Transform* representing BWT(*Text*).

Constraints

- The length of *Text* will be between 1 and 10^3.

Test Cases

Case 1

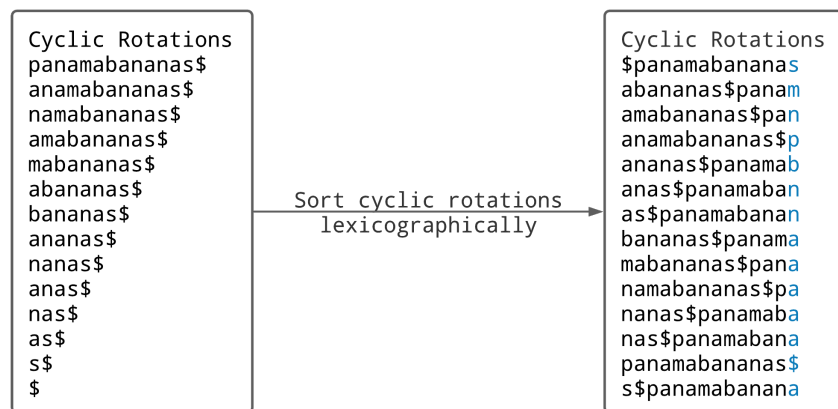**Description:** The sample dataset is not actually run on your code.
Input:

```
panamabananas
```

Output:

```
smnpbnnaaaaa$a
```

Figure:

```
Cyclic Rotations
panamabananas$
anamabananas$
namabananas$
amabananas$
mabananas$
abananas$
bananas$
ananas$
nanas$
anas$
nas$
as$
s$
$
```

Sort cyclic rotations
lexicographically

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

Shown above is a general (and inefficient) construction process for the Burrows-Wheeler Transform for the string `panamabananas`. We first generate all cyclic rotations of *Text* before sorting them lexicographically to build a Burrows-Wheeler matrix. Lastly, we output the last column of the Burrows-Wheeler matrix as the Burrows-Wheeler Transform of *Text*. This process is very similar to the process of generating a suffix array.

Case 2

Description: There are repeats in *Text*.

Input:
AATCAATC

Output:
CC$AATTAA

Case 3

Description: *Text* is made up of only one character.

Input:
AAAAAAAAAA$

Output:
AAAAAAAAAA$

Case 4

Description: *Text* is palindromic or has substrings that are palindromic.

Input:
GAGCAT$

Output:
TGCG$AA

Case 5

Description: A larger dataset of the same size as that provided by the randomized autograder. Check input/output folders for this dataset.