Burrows-Wheeler Transform (BWT) [TR-1994]

6.897
L13.4
Apr. 9, 2003

1. Append $ to end of string
2. Sort all n rotations of string in reverse order
   - same order as suffixes of reverse(string)
   ⇒ can compute in linear time
3. Write down 1ST letter in each rotation in order.

INTERESTING CLUSTERING!

$	n	a	n	a	$	b	a
n	a	$	b	a	n	a	
$	b	a	n	a	n	a	
a	n	a	n	a	$	b	
a	n	a	$	b	a	n	
a	$	b	a	n	a	n	
b	a	n	a	n	a	$	

put common contexts (prev. letters) together

BWT = nn$aaab

Can reconstruct string from BWT:

(A) every column is a permutation of string
(B) ⇒ last column = sort(first column) = sort(BWT).
(C) successor of (row in last column) = (row in first column)
(D) ⇒ first char. of string = (last row, first column)
(E) ith occurrence of char. in first col. ≡ ith occurrence of char. in last col.
(because of lexical sorting)

⇒ can unwind 1ST char, 2ND char, ...
in O(n) time using hashing.