CO202 – Software Engineering – Algorithms

String Matching - Solutions

Exercise 1: Prefix Function - Example 3

Complete the prefix function table

$$\pi[q] = \max\{k : k < q \text{ and } P_k \sqsupset P_q\}$$

i	1	2	3	4	5	6	7	8	9	10	11
$P[i]$	a	b	r	a	c	a	d	a	b	r	a
$\pi[i]$											

Exercise 1: Prefix Function - Example 3

Complete the prefix function table

$$\pi[q] = \max\{k : k < q \text{ and } P_k \sqsupset P_q\}$$

i	1	2	3	4	5	6	7	8	9	10	11
$P[i]$	a	b	r	a	c	a	d	a	b	r	a
$\pi[i]$	0	0	0	1	0	1	0	1	2	3	4

Exercise 2: Boyer-Moore Preprocessing

BCR Table

c	a	b	c	d	r
$bcr[c]$					

GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
$P[i]$	a	b	r	a	c	a	d	a	b	r	a
$gsr[i]$											

Exercise 2: Boyer-Moore Preprocessing

BCR Table

c	a	b	c	d	r
$bcr[c]$	11	9	5	7	10

GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
$P[i]$	a	b	r	a	c	a	d	a	b	r	a
$gsr[i]$											

Exercise 2: Boyer-Moore Preprocessing

BCR Table

c	a	b	c	d	r
$bcr[c]$	11	9	5	7	10

GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
$P[i]$	a	b	r	a	c	a	d	a	b	r	a
$gsr[i]$											1

Exercise 2: Boyer-Moore Preprocessing

BCR Table

c	a	b	c	d	r
$bcr[c]$	11	9	5	7	10

GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
$P[i]$	a	b	r	a	c	a	d	a	b	r	a
$gsr[i]$										3	1

Exercise 2: Boyer-Moore Preprocessing

BCR Table

c	a	b	c	d	r
$bcr[c]$	11	9	5	7	10

GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
$P[i]$	a	b	r	a	c	a	d	a	b	r	a
$gsr[i]$									10	3	1

Exercise 2: Boyer-Moore Preprocessing

BCR Table

c	a	b	c	d	r
$bcr[c]$	11	9	5	7	10

GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
$P[i]$	a	b	r	a	c	a	d	a	b	r	a
$gsr[i]$								10	10	3	1

Exercise 2: Boyer-Moore Preprocessing

BCR Table

c	a	b	c	d	r
$bcr[c]$	11	9	5	7	10

GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
$P[i]$	a	b	r	a	c	a	d	a	b	r	a
$gsr[i]$							7	10	10	3	1

Exercise 2: Boyer-Moore Preprocessing

BCR Table

c	a	b	c	d	r
$bcr[c]$	11	9	5	7	10

GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
$P[i]$	a	b	r	a	c	a	d	a	b	r	a
$gsr[i]$						7	7	10	10	3	1

Exercise 2: Boyer-Moore Preprocessing

BCR Table

c	a	b	c	d	r
$bcr[c]$	11	9	5	7	10

GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
$P[i]$	a	b	r	a	c	a	d	a	b	r	a
$gsr[i]$					7	7	7	10	10	3	1

Exercise 2: Boyer-Moore Preprocessing

BCR Table

c	a	b	c	d	r
$bcr[c]$	11	9	5	7	10

GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
$P[i]$	a	b	r	a	c	a	d	a	b	r	a
$gsr[i]$				7	7	7	7	10	10	3	1

Exercise 2: Boyer-Moore Preprocessing

BCR Table

c	a	b	c	d	r
$bcr[c]$	11	9	5	7	10

GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
$P[i]$	a	b	r	a	c	a	d	a	b	r	a
$gsr[i]$			7	7	7	7	7	10	10	3	1

Exercise 2: Boyer-Moore Preprocessing

BCR Table

c	a	b	c	d	r
$bcr[c]$	11	9	5	7	10

GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
$P[i]$	a	b	r	a	c	a	d	a	b	r	a
$gsr[i]$		7	7	7	7	7	7	10	10	3	1

Exercise 2: Boyer-Moore Preprocessing

BCR Table

c	a	b	c	d	r
$bcr[c]$	11	9	5	7	10

GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
$P[i]$	a	b	r	a	c	a	d	a	b	r	a
$gsr[i]$	7	7	7	7	7	7	7	10	10	3	1

Exercise 3: Worst-Case of Boyer-Moore

Show that the worst-case running time of the `BM-MATCHER` algorithm is $O(nm)$.

Exercise 3: Worst-Case of Boyer-Moore

Show that the worst-case running time of the BM-MATCHER algorithm is $O(nm)$.

It is easy to see when both P and T contain only one unique character, e.g. $P = a^m$ and $T = a^n$.

Exercise 4: Prefix Function as String Matcher

1) Devise an algorithm PF-MATCHER that uses the prefix function directly to find all occurrences of P in T?

2) What is the running time complexity of this method?

Exercise 4: Prefix Function as String Matcher

1) Devise an algorithm PF-MATCHER that uses the prefix function directly to find all occurrences of P in T?

```
PF-MATCHER(T,P)
 1: n = T.length
 2: m = P.length
 3: π = PREFIX-FUNCTION(P + ‘#’ + T)
 4: for i = 2*m+1 to n+m+1
 5:     if π[i] == m
 6:         PRINT(i-2*m-1)
```

Exercise 4: Prefix Function as String Matcher

2) What is the running time complexity of this method?

See discussion on the running time of the prefix function which yields $O(m)$ with m being the length of the pattern.

So for `PF-MATCHER`, we get $O(m + n)$.