CO202 – Software Engineering – Algorithms String Matching - Solutions

Exercise 1: Prefix Function - Example 3

Complete the prefix function table

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

i	1	2	3	4	5	6	7	8	9	10	11
P[i]	а	Ь	r	а	U	а	đ	а	b	r	а
$\pi[i]$											

String Matching

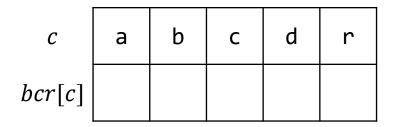
Exercise 1: Prefix Function - Example 3

Complete the prefix function table

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

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

BCR Table

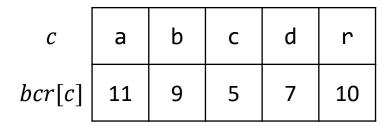


GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
P[i]	а	b	r	а	C	а	d	а	b	r	а
gsr[i]											

String Matching

BCR Table



GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
P[i]	а	b	r	а	C	а	d	а	b	r	а
gsr[i]											

BCR Table

С	а	b	С	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]	а	b	r	а	С	а	d	а	b	r	а
gsr[i]											1

BCR Table

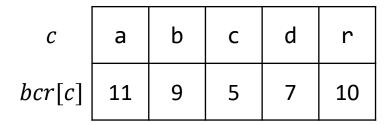
С	а	b	С	d	r
bcr[c]	11	9	5	7	10

GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
P[i]	а	b	r	а	C	а	d	а	р	٦	а
gsr[i]										3	1

String Matching

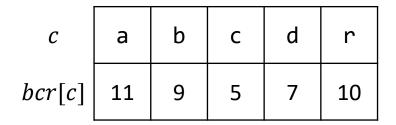
BCR Table



GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
P[i]	а	b	r	а	C	а	d	а	b	٦	a
gsr[i]									10	3	1

BCR Table

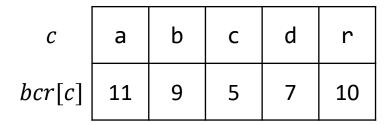


GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
P[i]	а	b	r	а	C	а	d	а	b	٦	a
gsr[i]								10	10	3	1

String Matching

BCR Table



GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
P[i]	a	b	r	а	C	а	d	а	b	٦	a
gsr[i]							7	10	10	3	1

String Matching

BCR Table

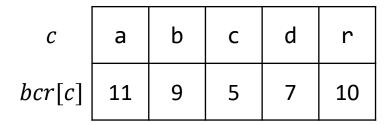
С	а	b	С	d	r
bcr[c]	11	9	5	7	10

GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
P[i]	а	b	r	а	U	а	d	а	р	r	а
gsr[i]						7	7	10	10	3	1

String Matching

BCR Table



GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
P[i]	a	b	r	а	C	а	d	а	b	٦	a
gsr[i]					7	7	7	10	10	3	1

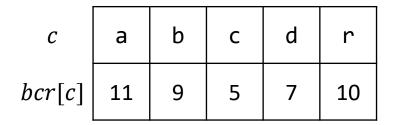
BCR Table

С	а	b	С	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	а	C	а	d	а	b	٦	а
gsr[i]				7	7	7	7	10	10	3	1

BCR Table



GSR Table

i	1	2	3	4	5	6	7	8	9	10	11
P[i]	a	b	r	а	С	а	d	а	b	۲	а
gsr[i]			7	7	7	7	7	10	10	3	1

String Matching

BCR Table

С	а	b	С	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]	а	b	r	а	C	а	d	а	b	r	а
gsr[i]		7	7	7	7	7	7	10	10	3	1

String Matching

BCR Table

С	а	b	С	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]	а	b	r	а	C	а	d	а	b	۲	а
gsr[i]	7	7	7	7	7	7	7	10	10	3	1

String Matching

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).