



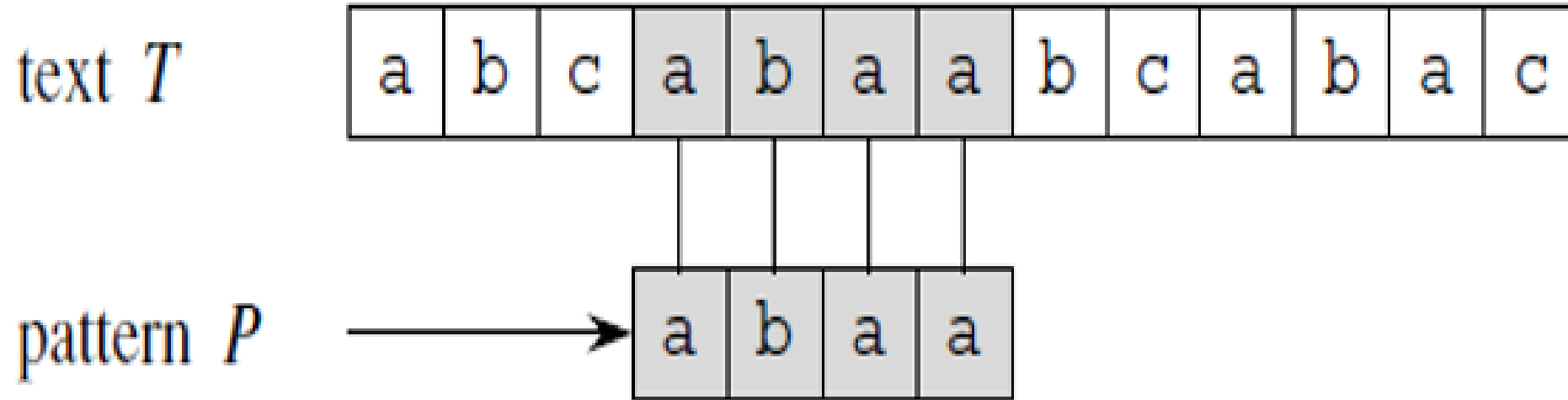
طراحی الگوریتم ها

فصل ۷: تطابق رشته ها (String Matching)

لیلا ناموری تازه کند

نیمسال دوم سال تحصیلی ۱۴۰۳-۱۴۰۴

✓ هدف مساله تطابق رشته ها این است که بررسی کند آیا رشته الگوی P بطول m در متن T بطول n موجود است یا نه؟ در صورت موجود، اندیس اولین مکان را برگرداند در غیراینصورت عدد -1 برگرداند.



روش های تطابق رشته ها

✓ روش ۱: هر کاراکتر رشته الگوی P با هر کاراکتر متن T مقایسه می شود و این مقایسه ها تا زمان پیدا شدن الگو و یا رسیدن به انتهای متن ادامه پیدا می کند.

```
int find(char *T, char *P)
```

```
{
```

```
    int t=0, p=0;
```

```
    int n=strlen(T), m=strlen(P);
```

```
    while(t < n && p < m)
```

```
    {
```

```
        if(T[t] == P[p])
```

```
        { t++; p++; }
```

```
        else if (p==0) t++;
```

```
        else { t -= (p-1); p=0; }
```

```
    }
```

```
    return(p==m?(t-m):-1);
```

```
}
```

T	c	c	d	b	a	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---	---

t

P	c	c	b
---	---	---	---

p

به انتهای متن و الگو نرسیده است؟

فعلا الگو و متن باهم یکسان هستند.

تطابق پیدا شده است.

پیچیدگی این الگوریتم در بدترین حالت برابر $O(nm)$ است که در آن n طول متن و m طول الگو می باشد.

✓مثال: الگوریتم قبلی را برای $T = \text{"ccdbaccbac"}$ و $P = \text{"ccb"}$ مرحله به مرحله اجرا کنید.

T

c	c	d	b	a	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---

t

P

c	c	b
---	---	---

p



$T[t] == P[p] \rightarrow t++, p++$

T

c	c	d	b	a	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---

t

P

c	c	b
---	---	---

p

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p



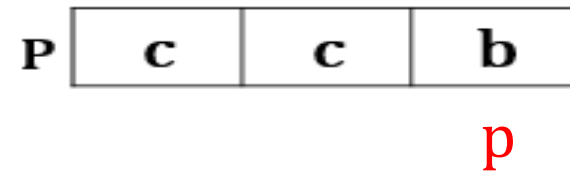
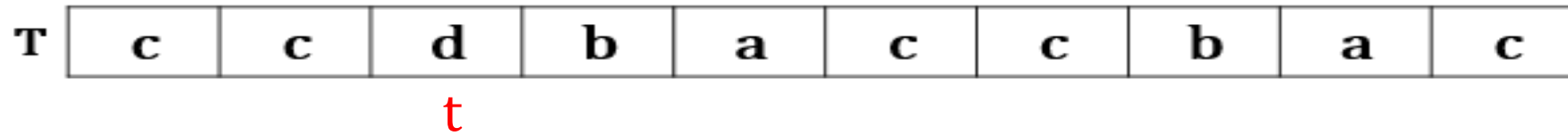
$T[t] == P[p] \rightarrow t++, p++$

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

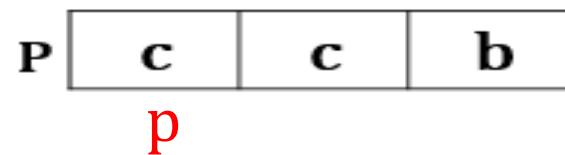
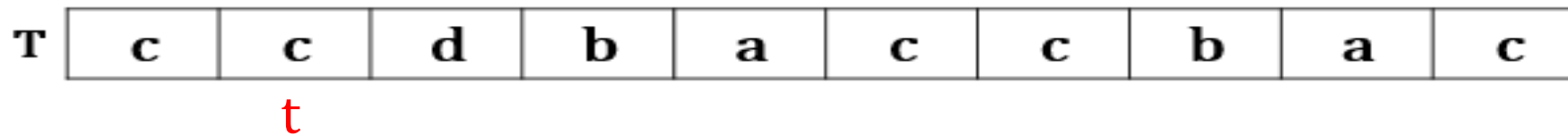
t

P	c	c	b
----------	----------	----------	----------

p



$T[t] \neq P[p] \ \&\& \ p \neq 0 \rightarrow t = (p-1); \ p = 0;$
 $t = (2-1) \rightarrow t = 1$



T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p



$T[t] == P[p] \rightarrow t++ ; p++$

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p



$T[t] \neq P[p] \ \&\& \ p \neq 0 \rightarrow t- = (p-1); \ p = 0;$
 $t- = (1-1) \rightarrow t=2$

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p



$T[t] \leftarrow P[p] \ \&\& \ p == 0 \rightarrow t++$

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p



$T[t] \neq P[p] \ \&\& \ p == 0 \rightarrow t++$

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p



$T[t] \neq P[p] \ \&\& \ p == 0 \rightarrow t++$

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p

T

c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P

c	c	b
----------	----------	----------

p



$T[t] == P[p] \rightarrow t++, p++$

T

c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P

c	c	b
----------	----------	----------

p

T

c	c	d	b	a	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---

t

P

c	c	b
---	---	---

p



$T[t] == P[p] \rightarrow t++, p++$

T

c	c	d	b	a	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---

t

P

c	c	b
---	---	---

p

T

c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P

c	c	b
----------	----------	----------

p



$T[t] == P[p] \rightarrow t++, p++$

T

c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P

c	c	b
----------	----------	----------

p

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

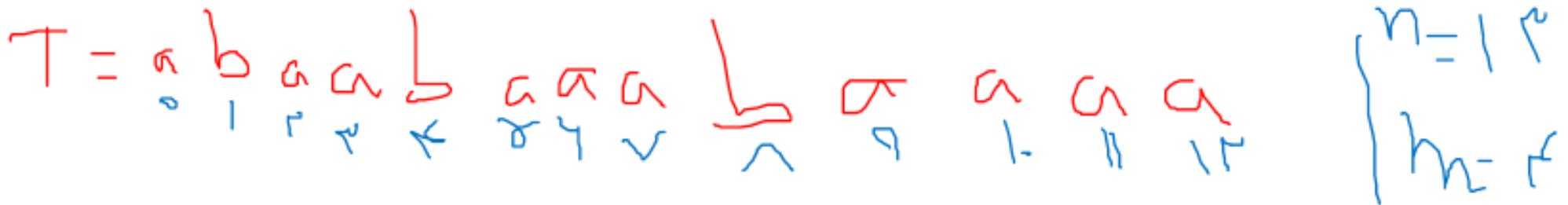
p



الگو پیدا شد
مقدار برگشتی برابر خواهد بود با
 $t-m=8-3=5$

✓مثال: الگوریتم قبلی را برای $T = \text{"abaabaaabaaaa"}$ و $P = \text{"aaaa"}$ مرحله به مرحله اجرا کنید.

جواب:

$T =$  $n=13$
 $h=4$

$P =$ 

t	0	1	1	2	3	4	3	4	4	5	6	7	8	6	7	8	7	8	8	9	10	11	12	13
p	0	1	0	0	1	2	0	1	0	0	1	2	3	0	1	2	0	1	0	0	1	2	3	4



الگو پیدا شد

مقدار برگشتی برابر خواهد بود با

$$t-m=13-4=9$$

✓ تمرین کلاسی: الگوریتم قبلی را برای $T = \text{"axabxabaxx"}$ و $P = \text{"aba"}$ مرحله به مرحله اجرا کنید.

✓ روش ۲ (روش Boyer-Moore): فرضاً n طول متن و m طول الگو باشد. در ابتدا، آخرین کاراکتر الگوی P (برعکس روش ۱ که از ابتدای الگو شروع می کرد) را با کاراکتر $T[m-1]$ مقایسه کرده در صورت یکسان بودن، داخل رشته ها را با هم مقایسه می کند. در غیر اینصورت همین کار را برای کاراکتر بعدی، رشته T انجام می دهد.

T

c	c	d	b	a	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---

t

P

c	c	b
---	---	---

p

$T[t] \neq P[p] \rightarrow t++$



T

c	c	d	b	a	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---

t

P

c	c	b
---	---	---

p

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p



$T[t] == P[p] \rightarrow t = t - 2, p = 0$

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p

T	c	c	d	b	a	c	c	b	a	c
----------	---	---	---	---	---	---	---	---	---	---

t

P	c	c	b
----------	---	---	---

p



$T[t] == P[p] \rightarrow t++, p++$

T	c	c	d	b	a	c	c	b	a	c
----------	---	---	---	---	---	---	---	---	---	---

t

P	c	c	b
----------	---	---	---

p

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p



$T[t] \neq P[p] \rightarrow t = 4, p = 2$

چون قبلا خانه سوم T را با آخرین عنصر P مقایسه کرده بود و الان نوبت خانه چهارم یعنی a است.

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p



$T[t] \leftarrow P[p] \rightarrow t++$

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p



$T[t] \leftarrow P[p] \rightarrow t++$

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p



$T[t] \leftarrow P[p] \rightarrow t++$

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p

T

c	c	d	b	a	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---

t

P

c	c	b
---	---	---

p



$T[t] == P[p] \rightarrow t = t - 2, p = 0$

T

c	c	d	b	a	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---

t

P

c	c	b
---	---	---

p

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p



$T[t] == P[p] \rightarrow t++, p++$

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p



$T[t] == P[p] \rightarrow t++, p++$

T	c	c	d	b	a	c	c	b	a	c
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

t

P	c	c	b
----------	----------	----------	----------

p

T

c	c	d	b	a	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---

t

P

c	c	b
---	---	---

p



$T[t] == P[p] \rightarrow t++, p++$

T

c	c	d	b	a	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---

t

P

c	c	b
---	---	---

p

T

c	c	d	b	a	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---

t

P

c	c	b
---	---	---

p



الگو پیدا شد
مقدار برگشتی برابر خواهد بود با
 $t-m=8-3=5$

پیچیدگی این الگوریتم در بدترین حالت برابر $O(nm)$ است که در آن n طول متن و m طول الگو می باشد.

✓ مثال: الگوریتم **Boyer-Moore** را برای $T = \text{"abaabaaabaaaa"}$ و $P = \text{"aaaa"}$ مرحله به مرحله اجرا کنید.

جواب:

$T = \text{"abaabaaabaaaa"}$ $n=14$
 $P = \text{"aaaa"}$ $m=4$

t	3	0	1	4	5	2	3	4	6	3	4	7	4	8	9	6	7	8	10	7	8	11	8	12	9
p	3	0	1	3	3	0	1	2	3	0	1	3	0	3	3	0	1	2	3	0	1	3	0	3	0



t	10	11	12	13
p	1	2	3	4

الگو پیدا شد
 مقدار برگشتی برابر خواهد بود با
 $t-m=13-4=9$

✓ تمرین کلاسی: الگوریتم **Boyer-Moore** را برای $T = \text{"axabxabaxx"}$ و $P = \text{"aba"}$ مرحله به مرحله اجرا کنید.

✓ روش ۳: KMP (Knuth-Morris-Pratt):

این الگوریتم رشته الگوی P بطول m را در متن T بطول n جستجو می کند و این کار را در زمان خطی $O(m+n)$ انجام می دهد.

در روش ۱، وقتی کاراکترهای $T[t]$ و $P[p]$ باهم یکسان نبودند (شکست) مقدار $p=0$ می شد یعنی بررسی را دوباره از ابتدا شروع می کرد (طبق قسمتی از الگوریتم روش یک)

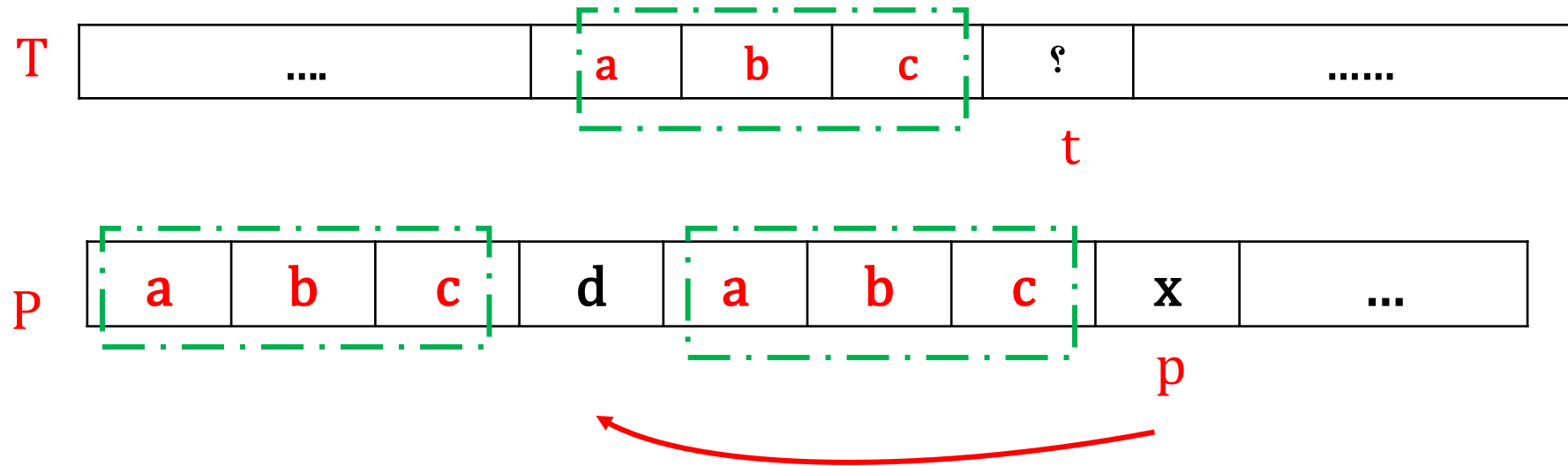
...

```
while(  $t < n$  &&  $p < m$  )  
{  
    if( $T[t] == P[p]$ )  
        {  $t++$ ;  $p++$ ; }  
    else if ( $p == 0$ )  $t++$ ;  
    else {  $t = (p-1)$ ;  $p = 0$ ; }  
}
```

.....

روش KMP می گوید که بعد از شکست، p نباید به ابتدای رشته الگو برگردد (یعنی $p=0$ شود) و همچنین در این شرایط t نیز نباید به عقب برگردد.

✓ **ب عنوان مثال:** فرضاً، در یک حالتی، قسمتی از متن T و رشته الگوی P بصورت زیر باشد:



✓ **چون** p نباید به خانه ص هستند.

T

....	a	b	c	?	
------	----------	----------	----------	----------	-------

 t

P

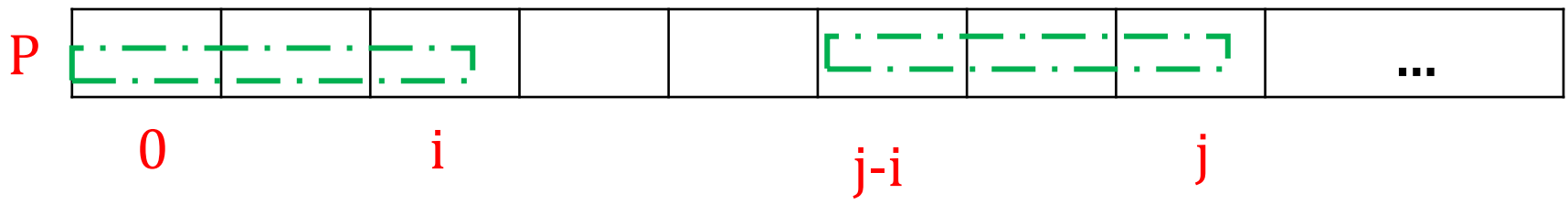
a	b	c	d	a	b	c	x	...
----------	----------	----------	----------	----------	----------	----------	----------	-----

 p

✓ **تعریف تابع شکست (failure):** اگر P رشته الگو باشد تابع شکست f بازای j بصورت زیر تعریف می شود:

$$f(j) = \begin{cases} \text{largest } i < j \text{ such that } P[0..i] = P[j-i \dots j] & \text{if such an } i \geq 0 \text{ exists} \\ -1 & \text{otherwise} \end{cases}$$

یعنی بزرگترین i ایی که در آن کلیه کاراکترهای صفر تا i برابر با کلیه کاراکترهای $j-i$ تا j باشند.



✓ **مثال:** تابع شکست را برای رشته الگوی $P=abcabcacab$ محاسبه کنید.

$j :$	0	1	2	3	4	5	6	7	8	9
$P :$	a	b	c	a	b	c	a	c	a	b
$f :$	-1	-1	-1	0	1	2	3	-1	0	1

✓ تابع شکست fail رشته الگوی P را دریافت کرده و مقادیر شکست f را در آرایه failure قرار می دهد.

✓ پیچیدگی این تابع برابر $O(m)$ می باشد.

✓ کار در کلاس: تابع شکست را برای رشته الگوی $P=ababababca$ محاسبه کنید.

j :	0	1	2	3	4	5	6	7	8	9
P:	a	b	a	b	a	b	a	b	c	a
f :	-1	-1	0	1	2	3	4	5	-1	0

✓ کار در کلاس: تابع شکست را برای رشته الگوی $P = abxabcabxabx$ محاسبه کنید.

کار در کلاس: تابع شکست را برای رشته الگوی $P = ababxababa$ محاسبه کنید.

```
int KMP(char *T, char *P)
{
    int t=0, p=0;
    int n=strlen(T), m=strlen(P);
    int[] failure=new int[m];
    fail(P, failure);
    while(t<n && p<m)
    {
        if(T[t]==P[p])
        { t++; p++; }
        else if (p==0) t++;
        else p=failure[p-1]+1;
    }
    return((p==m)?(t-m):-1) ;
}
```

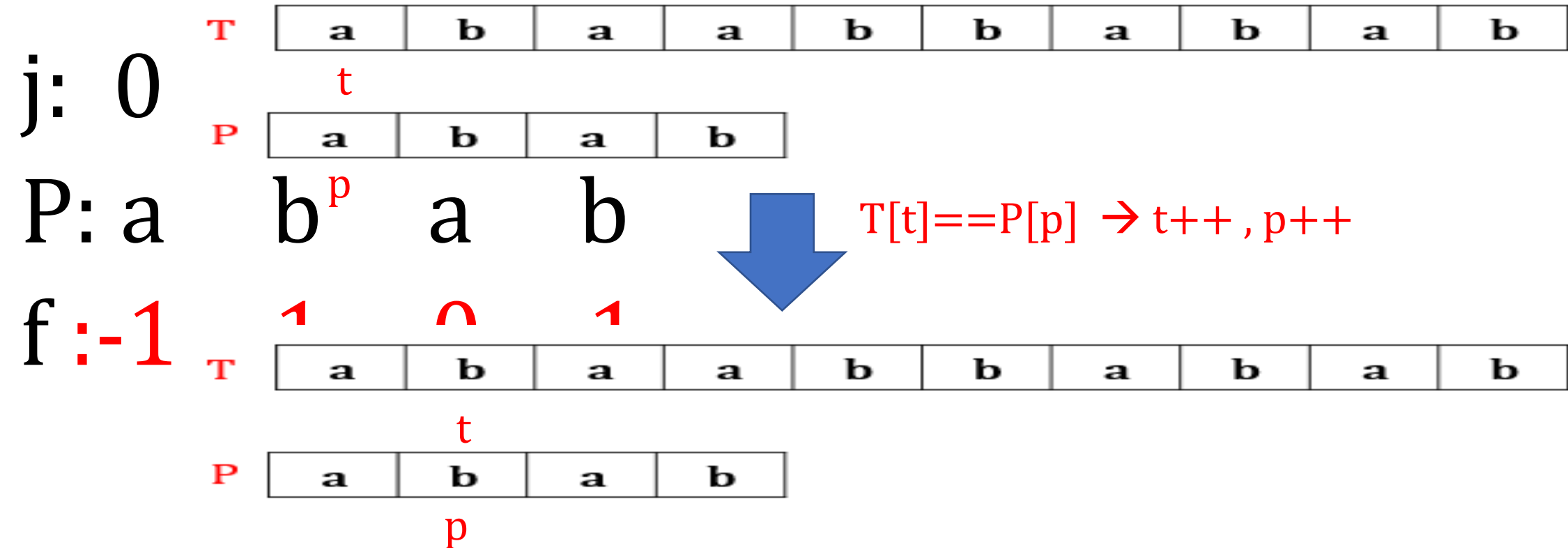
تابع fail مقادیر شکست را پیدا می کند و در آرایه failure قرار می دهد.

فعلا الگو و متن باهم یکسان هستند.

پیچیدگی این الگوریتم در بدترین حالت برابر $O(n+m)$ است که در آن n طول متن و m طول الگو می باشد.

✓ مثال: الگوریتم KMP را برای $T = \text{"abaabbabab"}$ و $P = \text{"abab"}$ مرحله به مرحله اجرا کنید.

جواب: ابتدا باید تابع شکست را برای الگوی P حساب کنیم:



j: 0 1 2 3

P: a

T

a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

t

P

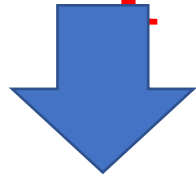
a	b	a	b
---	---	---	---

f: -1

-1

p

0



$T[t] == P[p] \rightarrow t++, p++$

T

a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

t

P

a	b	a	b
---	---	---	---

p

j: 0 1 2 3

P: a

f: -1

T

a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

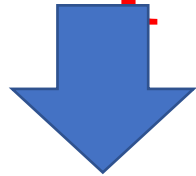
P

a	b	a	b
---	---	---	---

-1

0

1



$T[t] == P[p] \rightarrow t++, p++$

T

a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

P

a	b	a	b
---	---	---	---

p

j: 0 1 2 3

P: a

f: -1

T

a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

P

a	b	a	b
---	---	---	---

-1 0 1



$T[t] \neq P[p] \ \&\& \ p \neq 0 \rightarrow$
 $p = \text{failure}[p-1] + 1$
 $p = \text{failure}[3-1] + 1$
 $p = \text{failure}[2] + 1$
 $p = 0 + 1 = 1$

T

a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

P

a	b	a	b
---	---	---	---

p

j: 0 1 2 3

P: a

f: -1

T

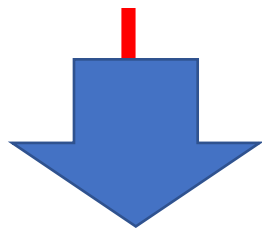
a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

P

a	b	a	b
---	---	---	---

-1

0



$T[t] \neq P[p] \ \&\& \ p \neq 0 \rightarrow$
 $p = \text{failure}[p-1] + 1$
 $p = \text{failure}[1-1] + 1$
 $p = \text{failure}[0] + 1$
 $p = -1 + 1 = 0$

T

a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

P

a	b	a	b
---	---	---	---

p

j: 0 1 2 3

P: a

f: -1

T

a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

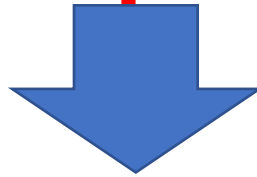
P

a	b	a	b
---	---	---	---

t

p

0



$T[t] == P[p] \rightarrow t++, p++$

T

a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

t

P

a	b	a	b
---	---	---	---

p

j: 0 1 2 3

P: a

f: -1

T

a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

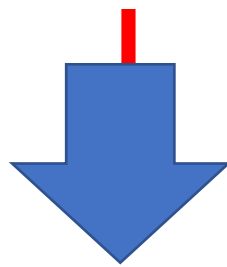
P

a	b	a	b
---	---	---	---

t

-1

0



$T[t] == P[p] \rightarrow t++, p++$

T

a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

P

a	b	a	b
---	---	---	---

t

p

j: 0 1 2 3

P: a

f: -1

T

a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

t

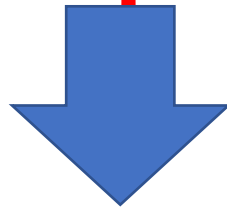
P

a	b	a	b
---	---	---	---

-1

0

p1



$T[t] \neq P[p] \ \&\& \ p \neq 0 \rightarrow$
 $p = \text{failure}[p-1] + 1$
 $p = \text{failure}[2-1] + 1$
 $p = \text{failure}[1] + 1$
 $p = -1 + 1 = 0$

T

a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

t

P

a	b	a	b
---	---	---	---

p

j: 0 1 2 3

P: a

f: -1

T

a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

P

a	b	a	b
---	---	---	---

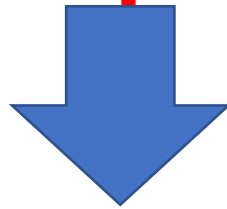
t

-1

p

0

1



$T[t] \neq P[p] \ \&\& \ p == 0 \rightarrow t++$

T

a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

P

a	b	a	b
---	---	---	---

t

p

j: 0 1 2 3

P: a

f: -1

T

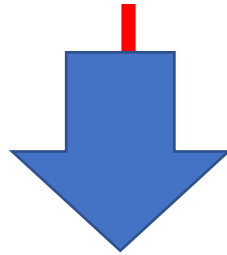
a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

t

P

a	b	a	b
---	---	---	---

-1 p U



$T[t] == P[p] \rightarrow t++, p++$

T

a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

t

P

a	b	a	b
---	---	---	---

p

j: 0 1 2 3

P: a

f: -1

T

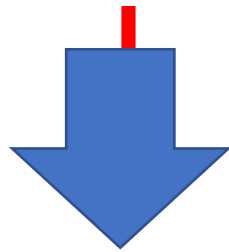
a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

P

a	b	a	b
---	---	---	---

-1

p



$T[t] == P[p] \rightarrow t++, p++$

T

a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

P

a	b	a	b
---	---	---	---

p

t

j: 0 1 2 3
P: a b a b
f: -1 -1 0 1

T

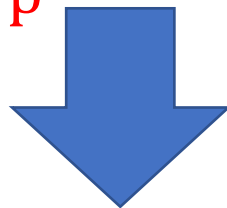
a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

P

a	b	a	b
---	---	---	---

t

p



$T[t] == P[p] \rightarrow t++, p++$

T

a	b	a	a	b	b	a	b	a	b
---	---	---	---	---	---	---	---	---	---

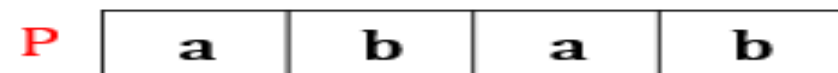
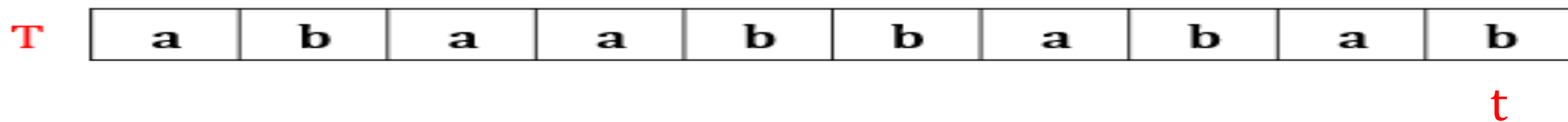
P

a	b	a	b
---	---	---	---

t

p

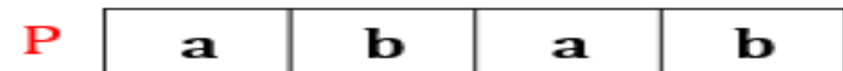
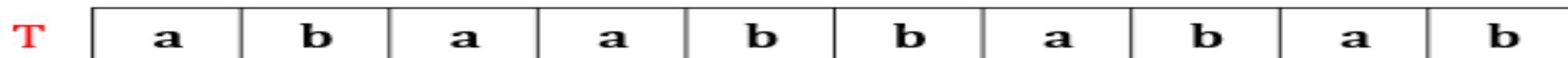
j: 0 1 2 3
P: a b a b
f: -1 -1 0 1



p



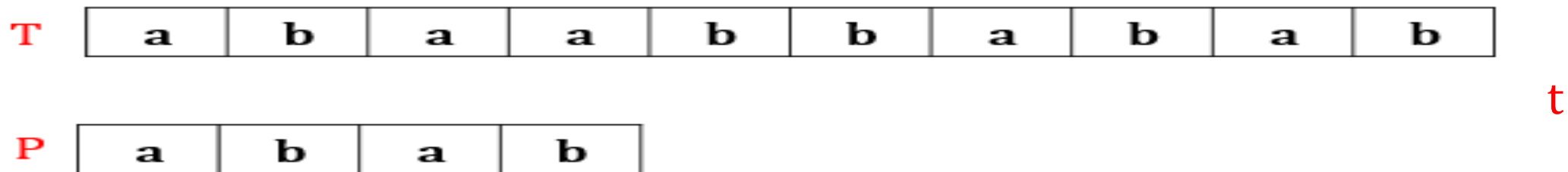
$T[t] == P[p] \rightarrow t++, p++$



t

p

j: 0 1 2 3
P: a b a b
f: -1 -1 0 1



p



شرط خروج از حلقه درست می شود: $t == n \ \&\& \ p == m$

`return((p==m)?(t-m):-1) ;`
`return (10-4=6);`

تمرینات سری ۱

۱- تابع شکست را برای رشته الگوی $P=abcbabcba$ محاسبه کنید.

۲- الگوریتم KMP را برای $T="abcababcabc"$ و $P="abcabc"$ مرحله به مرحله اجرا کنید.

پایان فصل ۷