

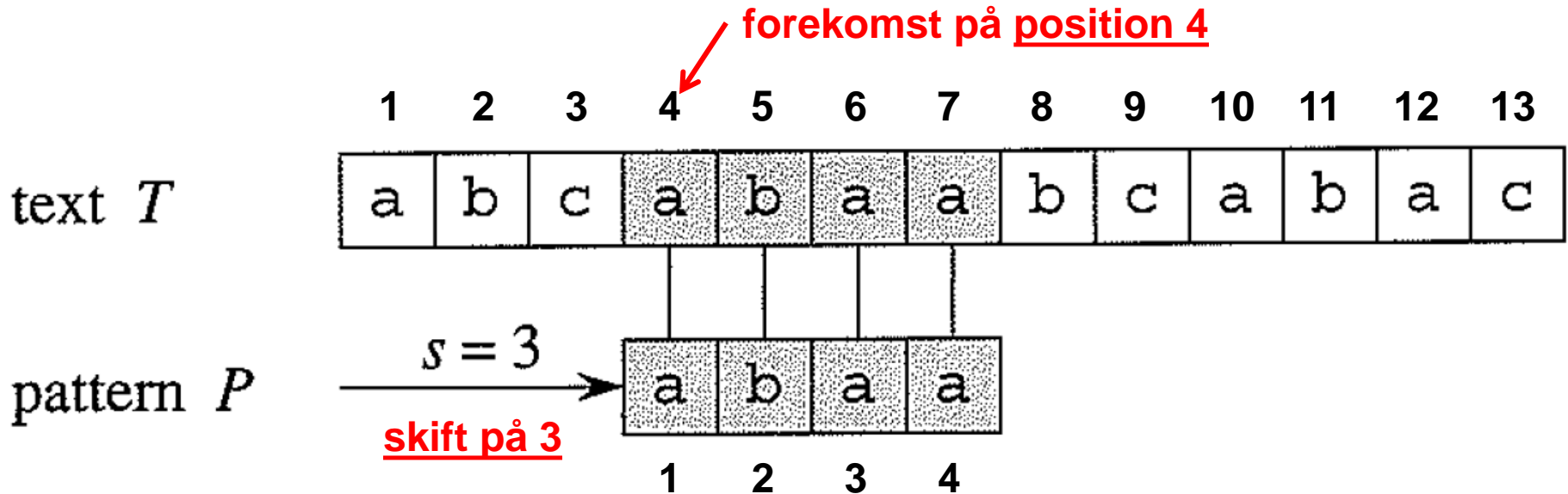
# Algoritmer og Datastrukturer 2

Gerth Stølting Brodal

Mønstergenkendelse [CLRS, kapitel 32.1-32.2, 32.4]



# Mønster genkendelse



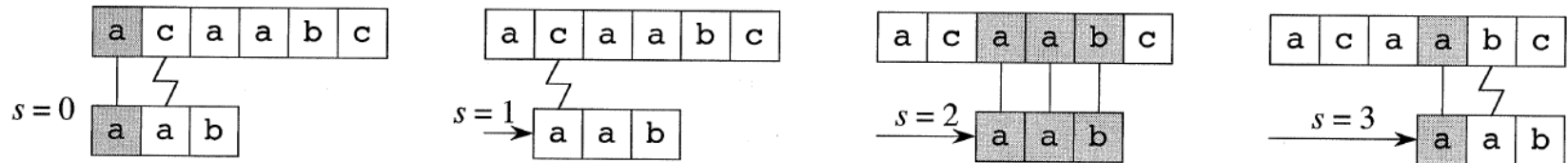
**Input:** Tekst  $T$  af længde  $n$  og mønster  $P$  af længde  $m$

**Output:** Alle positioner i  $T$  hvor  $P$  forekommer

# Naive Algorithm

NAIVE-STRING-MATCHER( $T, P$ )

- 1  $n = T.length$
- 2  $m = P.length$
- 3 **for**  $s = 0$  **to**  $n - m$
- 4     **if**  $P[1..m] == T[s + 1..s + m]$
- 5         **print** “Pattern occurs with shift”  $s$



$O(n \cdot m)$

# Rabin-Karp : Eksempel $P = 31415$

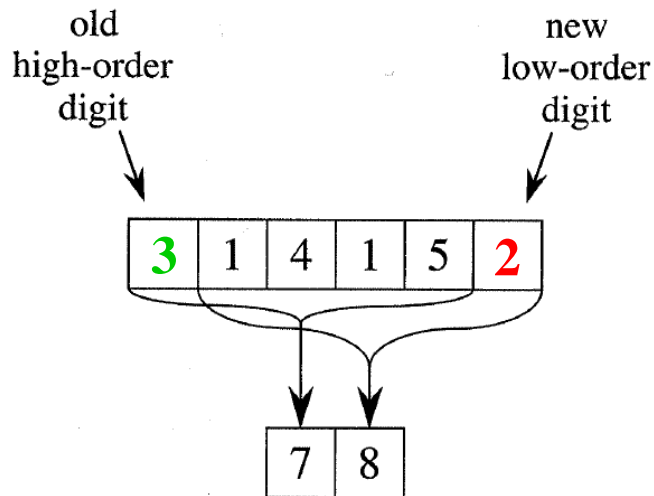
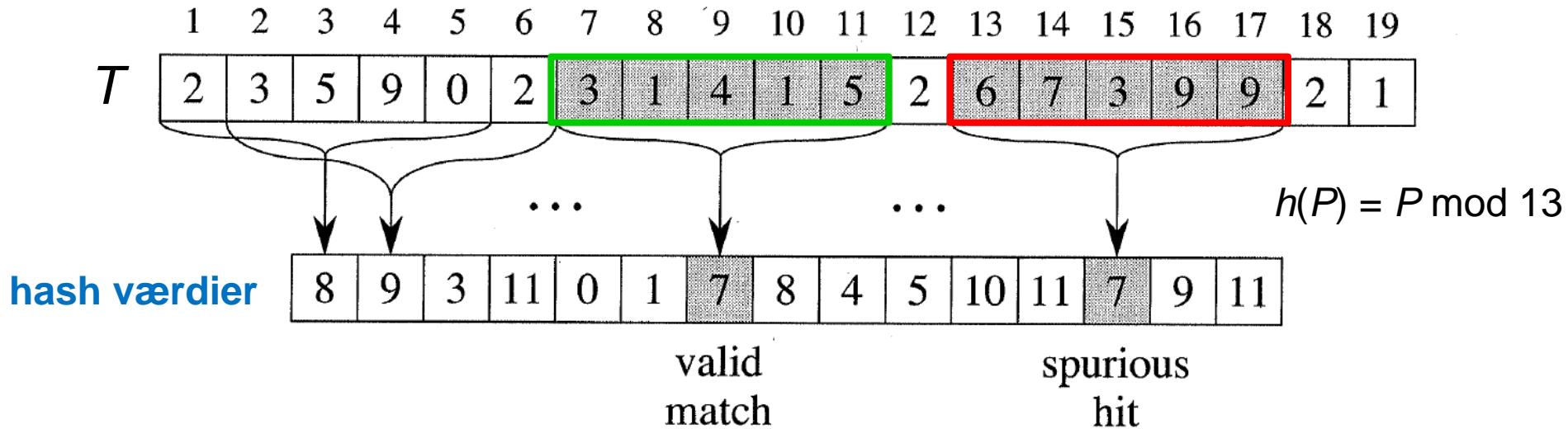


Diagram illustrating the rolling hash update process:

old high-order digit: 3 (green)

new low-order digit: 2 (red)

shift

$$14152 \equiv (31415 - 3 \cdot 10000) \cdot 10 + 2 \pmod{13}$$

$$\equiv (7 - 3 \cdot 3) \cdot 10 + 2 \pmod{13}$$

$$\equiv 8 \pmod{13}$$

$(a \cdot b) \bmod p = ((a \bmod p) \cdot b) \bmod p$        $(a+b) \bmod p = ((a \bmod p) + b) \bmod p$   
 $(a+p \cdot x) \bmod p = a \bmod p$ ,    f.eks.  $24 \bmod 13 = 11 = -2 \bmod 13$

# Rabin-Karp

RABIN-KARP-MATCHER( $T, P, d, q$ )

```

1   $n = T.length$ 
2   $m = P.length$ 
3   $h = d^{m-1} \bmod q$ 
4   $p = 0$ 
5   $t_0 = 0$ 
6  for  $i = 1$  to  $m$                                 // preprocessing
7       $p = (dp + P[i]) \bmod q$ 
8       $t_0 = (dt_0 + T[i]) \bmod q$ 
9  for  $s = 0$  to  $n - m$                                 // matching
10     if  $p == t_s$ 
11         if  $P[1..m] == T[s + 1..s + m]$ 
12             print "Pattern occurs with shift"  $s$ 
13     if  $s < n - m$ 
14          $t_{s+1} = (d(t_s - T[s+1] \cdot h) + T[s + m + 1]) \bmod q$ 

```

$$p = P[1]d^{m-1} + P[2]d^{m-2} + \dots + P[m-1]d^1 + P[m]d^0 \bmod q$$

$O(n \cdot m)$

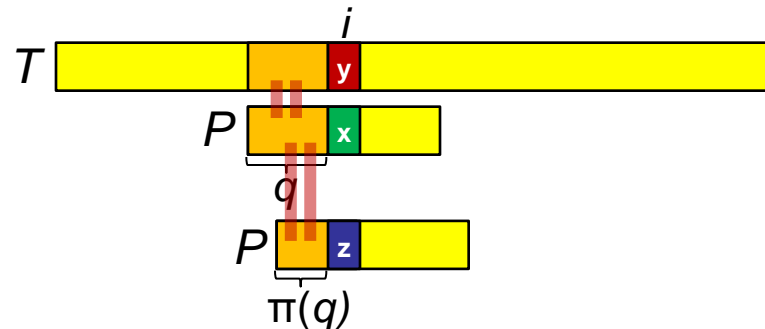
# Knuth-Morris-Pratt

1977

KMP-MATCHER( $T, P$ )

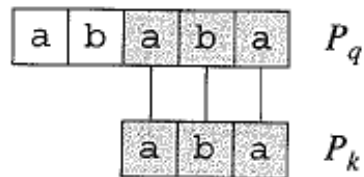
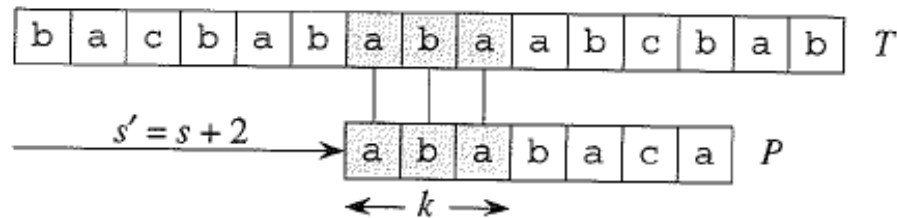
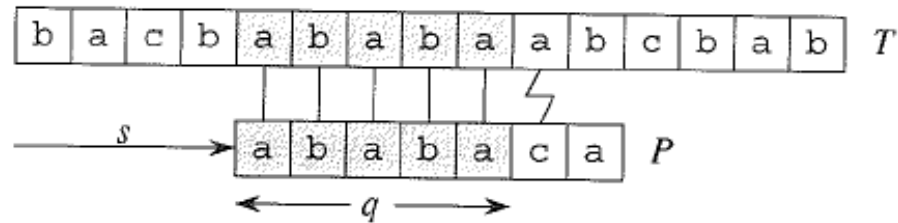
```
1   $n = T.length$ 
2   $m = P.length$ 
3   $\pi = \text{COMPUTE-PREFIX-FUNCTION}(P)$ 
4   $q = 0$  // number of characters matched
5  for  $i = 1$  to  $n$  // scan the text from left to right
6    while  $q > 0$  and  $P[q + 1] \neq T[i]$ 
7       $q = \pi[q]$  // next character does not match
8    if  $P[q + 1] == T[i]$ 
9       $q = q + 1$  // next character matches
10   if  $q == m$  // is all of  $P$  matched?
11     print "Pattern occurs with shift"  $i - m$ 
12      $q = \pi[q]$  // look for the next match
```

$\pi(0) = 0$   
 $\pi(q) = \max \{ i \mid i < q \text{ og } P[1..i] \text{ er et suffix af } P[1..q] \}$



$O(n)$

# Knuth-Morris-Pratt: Eksempel



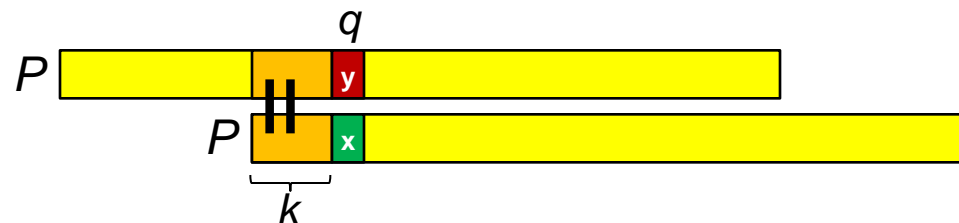
$$\pi(0) = 0$$

$$\pi(q) = \max \{ i \mid i < q \text{ og } P[1..i] \text{ er et suffix af } P[1..q] \}$$

# Knuth-Morris-Pratt: Beregning af prefix funktionen

COMPUTE-PREFIX-FUNCTION( $P$ )

```
1   $m = P.length$ 
2  let  $\pi[1..m]$  be a new array
3   $\pi[1] = 0$ 
4   $k = 0$ 
5  for  $q = 2$  to  $m$ 
6      while  $k > 0$  and  $P[k + 1] \neq P[q]$ 
7           $k = \pi[k]$ 
8      if  $P[k + 1] == P[q]$ 
9           $k = k + 1$ 
10      $\pi[q] = k$ 
11 return  $\pi$ 
```

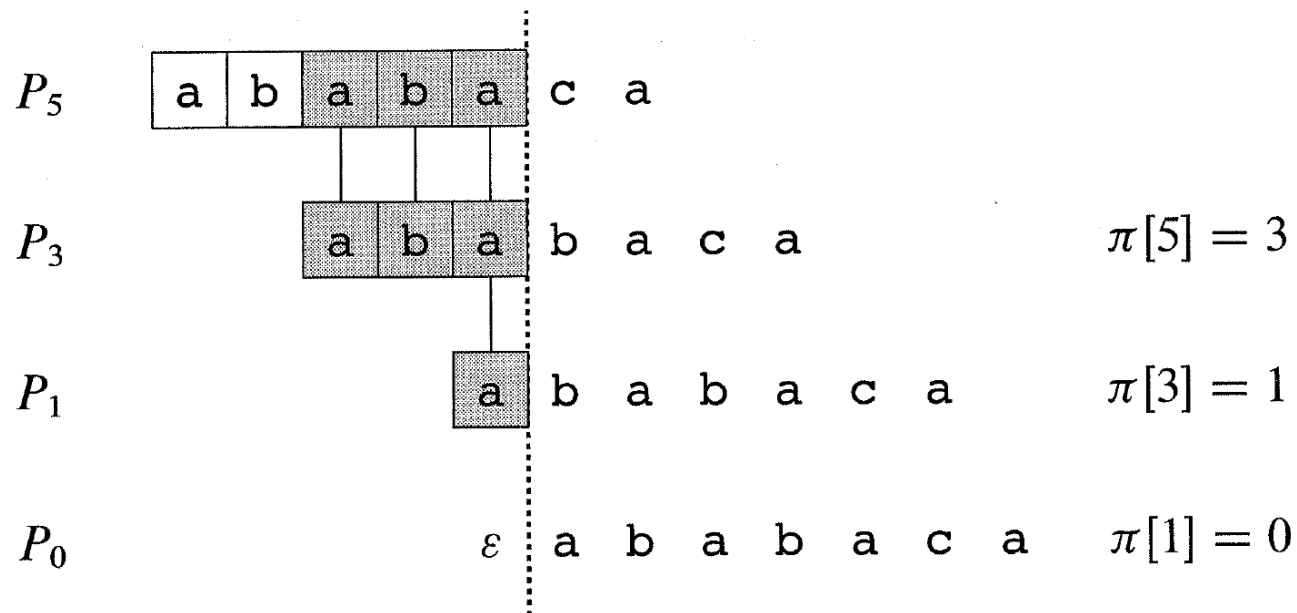


$O(m)$



# Knuth-Morris-Pratt: Beregning af prefix funktionen

$i$	1	2	3	4	5	6	7
$P[i]$	a	b	a	b	a	c	a
$\pi[i]$	0	0	1	2	3	0	1



# Worst-case tider

Algorithm	Preprocessing time	Matching time	[CLRS]
Naive	0	$O((n - m + 1)m)$	32.1
Rabin-Karp	$\Theta(m)$	$O((n - m + 1)m)$	32.2
Finite automaton	$O(m  \Sigma )$	$\Theta(n)$	(32.3)
Knuth-Morris-Pratt	$\Theta(m)$	$\Theta(n)$	32.4