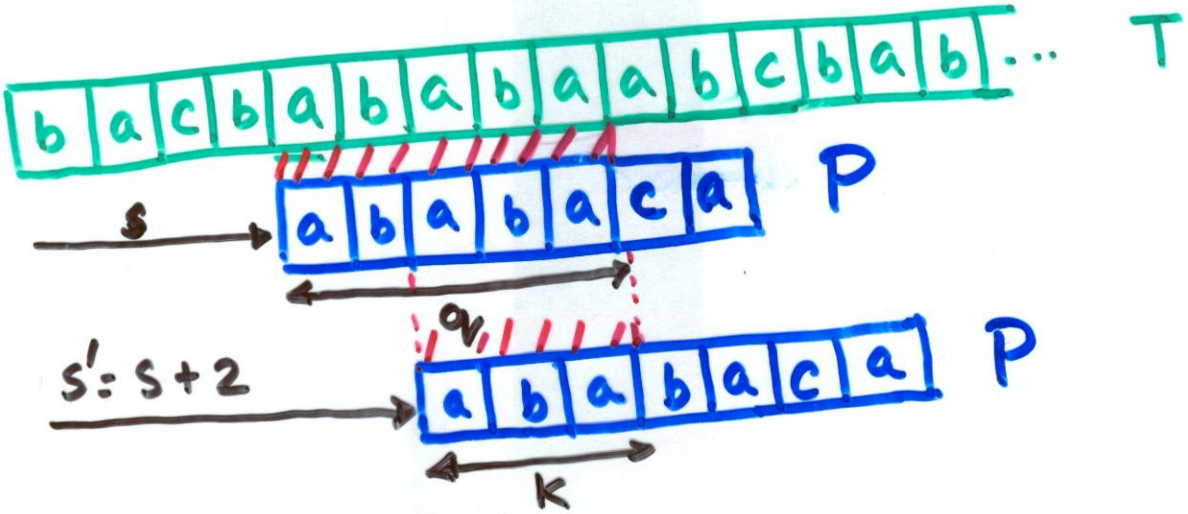
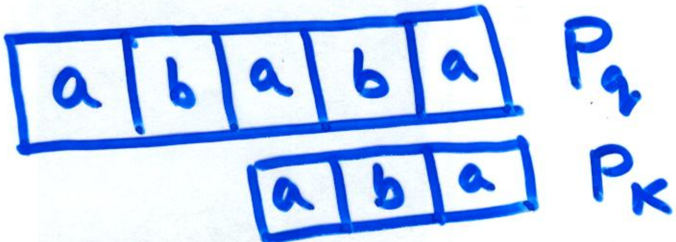
Knuth-Morris-Pratt algorithm

It turns out that preprocessing of P can be done in $O(m)$ time, and matching still taking $O(n)$ time. KMP algorithm thus matches strings in $\Theta(m+n)$ time.

Prefix function π





Given $P[1 \ldots q]$ matches text $T[s+1 \ldots s+q]$ what is the least $s' > s$ s.t.

$P[1 \ldots k] = T[s'+1 \ldots s'+k]$ for $s'+k = s+q$?

Equivalently, we ask:

what is the largest $k < q$ s.t. $P_k \sqsupset P_q$?

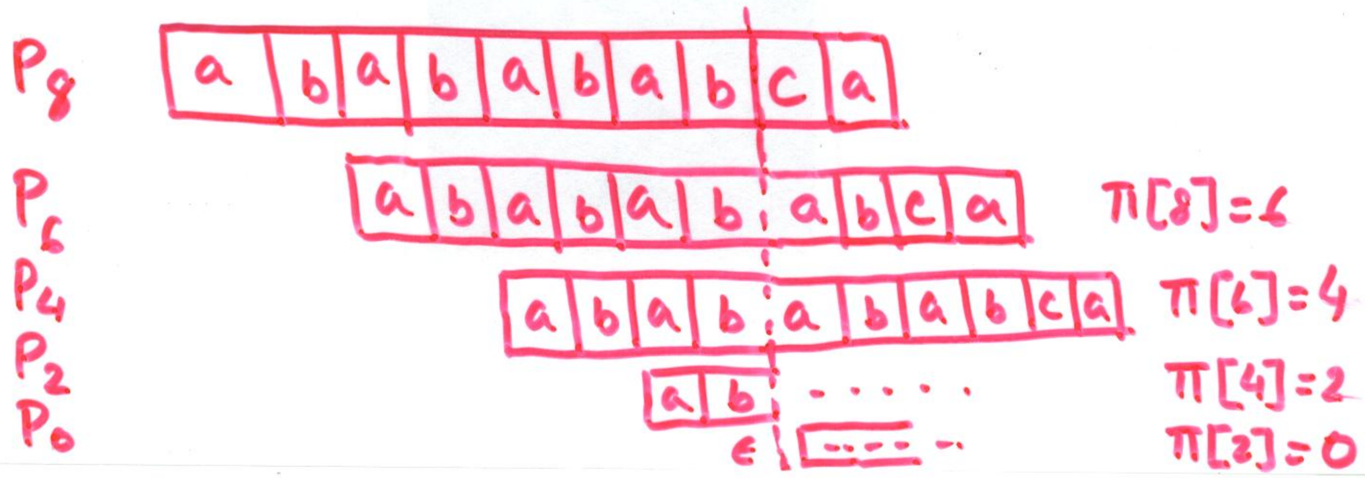
Then, $s' = s + (q-k)$ is next potentially valid shift.

Prefix function $\pi : \{1, 2, \ldots, m\} \to \{0, 1, \ldots, m-1\}$

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

$\pi[q]$ is the length of the longest prefix of P that is a proper suffix of P_q.

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



```
KMP-Match(T, P)
n := length(T)
m := length(P)
π := Prefix(P)
q := 0
for i := 1 to n
    while q > 0 and P[q+1] ≠ T[i]
①  - - - -      q := π[q];
    if P[q+1] = T[i]
②  . . . .   then q := q+1;
    if q = m
       then print "match with shift i-m".
③  - - - -      q := π[q];
```

Time analysis:

- q is always non-negative.
- It decreases in ① and ③.
- While loop can not have complexity more than the decrease in q.
- Total increase in q is $O(n)$ in the for loop which are decreased in ① and ③.
- So, total complexity is $O(n)$.

```
Prefix (P)
  m := length [P]
  π[1] := 0
  k := 0
  for q := 2 to m
      while k>0 and P[k+1] ≠ P[q]
            k := π[k];
      if P[k+1] = P[q] then
            k := k+1;
      π[q] := k;
  return π
```

Time analysis: Similar as before.
It is $O(m)$.

P:

a	b	a	b	a	b	a	b	c	a

$q=2 \rightarrow \pi[2]=0$

$q=3 \rightarrow k=1,\ \pi[3]=1$

$q=4 \rightarrow P[2]=P[4] \rightarrow k=2 \rightarrow \pi[4]=2$

$q=5 \rightarrow P[3]=P[5] \rightarrow k=3 \rightarrow \pi[5]=3$

$q=6 \rightarrow P[4]=P[6] \rightarrow k=4 \rightarrow \pi[6]=4$

P: | a | b | a | b | a | b | a | b | c | a |

$q=7 \to P[5] = P[7] \to k=5 \to \pi[7]=5$

$q=8 \to P[6] = P[8] \to k=6 \to \pi[8]=6$

$q=9 \to P[7] \neq P[9] \to k=\pi[6]=4$

$P[5] \neq P[9] \to k=\pi[4]=2$

$P[3] \neq P[9] \to k=\pi[2]=0$

$P[1] \neq P[9] \to \pi[9]=0$

$q=10 \to P[1] = P[10] \to k=1,\ \pi[10]=1.$

Correctness of the KMP algorithm needs a proof. See the book.