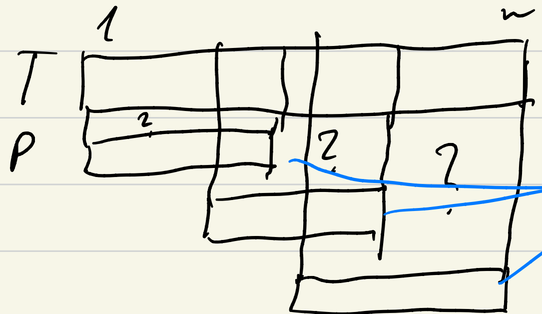
SEARCHING STRINGS $\Sigma = \{\sigma_1, \dots, \sigma_d\}$

$T/1 : \Sigma[n]$ text || pattern: $P/1 : \Sigma[m]$ || $n, m \in \mathbb{N}_0$

$n \geq m > 0$



different positions of P

$P[1..m] \stackrel{?}{=} T[s+1..s+m]$

Def $s \in 0..n-m$ is a valid shift of P on T, iff

$$T[s+1..s+m] = P[1..m]$$

We search for the set of valid shifts:

$$S = \{ s \in 0..(n-m) \mid T[s+1..s+m] = P[1..m] \}$$

PATTERN MATCHING

STRING MATCHING

Example: $P[1..4] = BABA$

$$
\begin{array}{lccccccccccc}
 & 1 & & & 4 & & & & 8 & & & 11 \\
T[1..11] = & A & B & A & B & B & A & B & A & B & A & B \\
P[1..4] = & \cancel{B} & A & B & A & & & & & & & \\
 & & B & A & B & \cancel{A} & & & & & & \\
 & & & \cancel{B} & A & B & A & & & & & \\
 & & & & B & \cancel{A} & B & A & & & & \\
 & & & & & B & A & B & A & & & \\
 & & & & & & \cancel{B} & A & B & A & & \\
 & & & & & & & B & A & B & A & \\
 & & & & & & & & \cancel{B} & A & B & A
\end{array}
$$

$P = AAAA$
$T = AA \dots A$

$s = 4$

$s = 6$

$S = \{4, 6\}$

$P \wedge T = \emptyset$

BRUTE-FORCE:
$mT(n,m) \in \Theta(n-m+1)$
$MT(n,m) \in \Theta((n-m+1) * m)$

$mT(m) \in \Theta(1)$
$MT(m) \in \Theta(m)$

BruteForce($T/1{:}\Sigma[n]$; $P/1{:}\Sigma[m]$; $S : \mathbb{N}\{\}$)

$S := \{\}$
$s := 0$ to $n - m$
$T[s+1..s+m] = P[1..m]$
$S := S \cup \{s\}$ | SKIP

$(T[s+1..s+m] = P[1..m]) : \mathbb{B}$

$j := 1$
$j \le m \wedge T[s+j] = P[j]$
$j{+}{+}$
return $j > m$

If m is relatively small compared to: (practical case)

$mT_{BF}(n,m) \in \Theta(n)$

$MT_{BF}(n,m) \in \Theta(n * m)$

Quicksearch

Example: $P[1..4] = CADA \quad || \quad m = 4 \quad || \quad \Sigma = \{A, B, C, D\}$

```
           CADA              CADA                CADA            CADA
Text: ...xxxxA.......xxxxB..........xxxxC........xxxxD..
            CADA                 CADA             CADA            CADA
```

If $T[s+m+1] = \sigma$, then we can make the following table:

σ	A	B	C	D
shift(σ)	1	5	4	2

$|\Sigma|$ is constant $\Rightarrow T_{iS}(m) \in \Theta(m)$

σ	A	B	C	D
initial shift(σ)	5	5	5	5
C			4	
A	3			
D				2
A	1			
final shift(σ)	1	5	4	2

initShift(P/1:Σ[m])

- $\forall \sigma \in \Sigma$
  - shift(σ) := m+1
- j := 1 to m
  - shift(P[j]) := m+1−j

QuickSearch(T/1:Σ[n]; P/1:Σ[m]; S:N{})

- initShift(P); S := {}; s := 0 — $\Theta(m)$
- s+m ≤ n — $\Theta((n-m+1)*m)$; $\Theta\left(\frac{n}{m+1}\right)$
  - T[s+1..s+m] = P[1..m]
    - S := S ∪ {s} | (skip)
  - s+m < n
    - s += shift(T[s+m+1]) | break

$MT_{QS}(n,m) \in \Theta((n-m+2)*m)$

$mT_{QS}(n,m) \in \Theta\left(\frac{n}{m+1}+m\right)$

$i =$ 4 8 12 16 20 23

$T[i] =$ | A D A B | A B C A | D A B C | A B A D | A C A D | A D A |

~~C~~ A D A

~~C~~ A D A

$s = 6$

C A D A

C A D A

~~C~~ A D A

$s = 17$

C A D A

~~C~~ A D A

$$\boxed{S = \{6, 17\}}$$

σ	A	B	C	D
$shift(\sigma)$	1	5	4	2

<u>Def</u> $x \sqsupset y$ (string x is suffix of string y)

<u>iff</u> $\exists z \neq$ empty: $z + x = y$

(x may be empty)

$\overbrace{\qquad\qquad\qquad}^{y}$ $\underbrace{\qquad\qquad}_{x}$

<u>Def</u> $x \sqsupseteq y$ <u>iff</u> $x \sqsupset y$ <u>or</u> $x = y$

KNUTH-MORRIS-PRATT (KMP) algorithm

$P[1..8] = BABABBAB$

```
i    =          4    6   8       12     15    18
T[i] = A B A B | A B A B | B A B A | B A B | B A B
       B̶
         B A B A B B̶
s=3          B A B A B B A B
                         B A B A B B̶
s=10                       B A B A B B A B
                                       B A B
```

$\boxed{S = \{3, 10\}}$

Def $next[j] = \max\{h \in 0..(j-1) \mid P_h \sqsupset P_j\}$

$P[j] =$	B	A	B	A	B	B	A	B
$j =$	1	2	3	4	5	6	7	8
$next[j] =$	0	0	1	2	3	1	2	3

Notation:

$P_j = P[1..j]$

(only for KMP alg.)

I. $0 \le next[j+1] \le next[j]+1$

$j > next[j] \ge 0$

$KMP(T/1:\Sigma[n];\ P/1:\Sigma[m];\ S:\mathbb{N}\{\})$

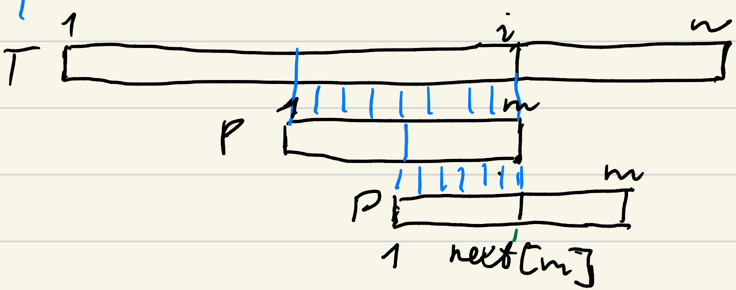
$next/1:\mathbb{N}[m];\ init(next, P);\ S:=\{\};\ i:=j:=0$

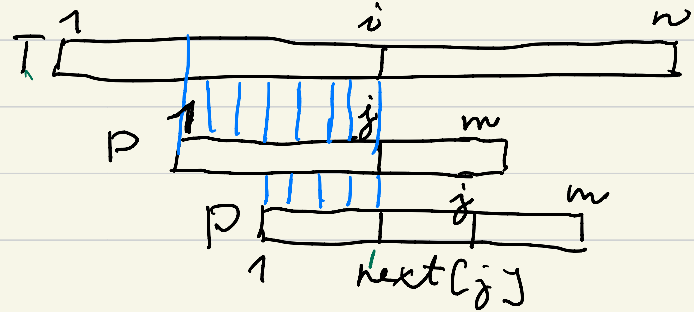
$P_j \sqsupseteq T_i$

$0 \le j \le i \le n$

$i < n$

- $T[i+1] = P[j+1]$
  - $i{+}{+};\ j{+}{+}$
  - $j = m$
    - $S := S \cup \{i-m\}$
    - $j := next[m]$
  - else: X
- else: $j = 0$
  - $i{+}{+}$
  - else: $j := next[j]$





$\underline{\text{Def}}$ $next[j] = \max\{h \in 0..(j-1) \mid P_h \sqsupset P_j\}$ $\quad j \in 1..m$

(Later: $mT_{init}(m), MT_{init}(m) \in \Theta(m)$) $\xrightarrow{m \le n}$ (*)

$t(i,j) \overset{def}{=} 2i - j \in 0..2n$

starts from 0, strictly increases on each branch of the loop, $t(i,j) \le 2n$

$\Downarrow$

The loop iterates max $2n$ times

The loop iterates min n times

($i=0$ in the beginning,
i increases max. 1 in 1 iteration
the loop stops when $i \ge n$ } $i = n$
Inv: $i \le n$)

$\Rightarrow mT_{loop}(n), MT_{loop}(n) \in \Theta(n)$

(*) $\boxed{mT_{KMP}(n), MT_{KMP}(n) \in \Theta(n)}$

init(next/1 : $\mathbb{N}[m]$; P/1 : $\Sigma[m]$)

- next[1] := 0; i := 0; j := 1
- **while** $j < m$
  - **if** $P[i+1] = P[j+1]$
    - **then:**
      - i++
      - j++
      - next[j] := i
    - **else:** **if** $i = 0$
      - **then:**
        - j++
        - next[j] := 0
      - **else:**
        - i := next[i]

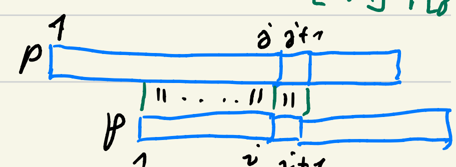
invariant: $0 \le i < j \le m \ \wedge$ $P_i \sqsupset P_j \ \wedge$ next[1..j] is already calculated.

Def: $next[j] = \max\{h \in 0..(j-1) \mid P_h \sqsupset P_j\}$

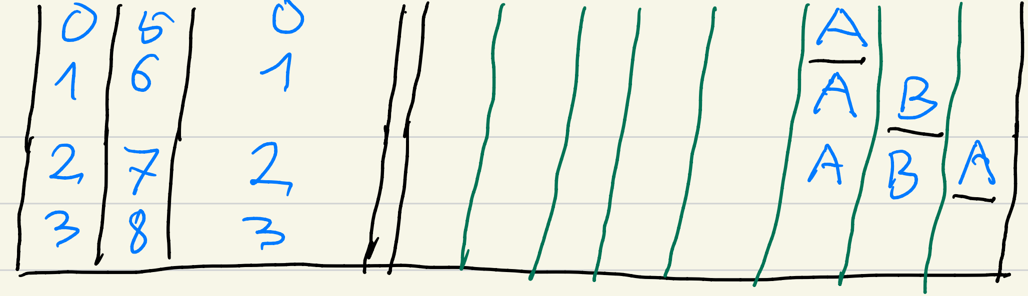
I. $next[j+1] \le next[j]+1$

Example:
$P[1..8] = ABABBABA$

I. $P_{i+1} \sqsupset P_{j+1} \Leftrightarrow P_i \sqsupset P_j \wedge P[i+1]=P[j+1]$



i	j	next[j]	A (1)	B (2)	A (3)	B (4)	B (5)	A (6)	B (7)	A (8)
0	1	0		~~A~~						
0	2	0			A					
1	3	1			A	B				
2	4	2			A	B	~~A~~			
0	4	2					~~A~~			

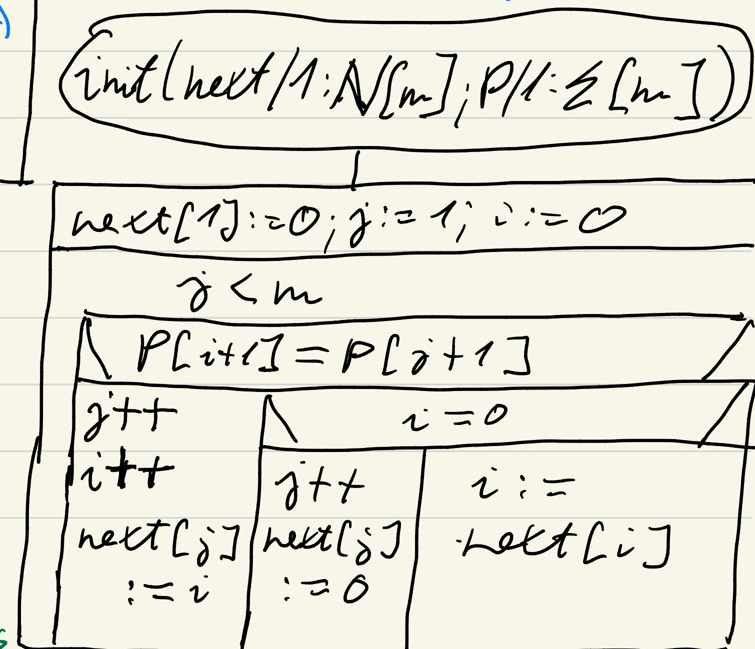


Result:

$0 \le i < j \le m$

$P[j] =$	A	B	A	B	B	A	B	A
$j =$	1	2	3	4	5	6	7	8
$next[j] =$	0	0	1	2	0	1	2	3

(it. ~ iteration)



Efficiency of init(next, P):

$t(j,i) := 2j - i \in 2..2m$

Before 1st it. of loop: $t(j,i) = 2$

$t(j,i)$ is strictly increasing in each it. of the loop.

$\Rightarrow$ the loop iterates max. $(2m-2)$ times.

$j = 1$ before 1st it.
j increases max. by 1 in 1 it.
$j \leq n$

the loop iterates min. $(n-1)$ times

$$\boxed{mT_{init}(n), MT_{init}(n) \in \Theta(n)}$$