Tutorial Week 3

Definition 1. *A string is a* **border** *of another string, if the two are different and the former occurs both as a prefix and as a suffix for the later. For a string w, an integer p with $0 < p \leq |w|$ is a* **period** *of w if w has a border of length $|w| - p$.*

Exercise 1. *Consider the following algorithm calculating the length of the longest borders of all prefixes of a string x of length m.*

Algorithm 1 Compute borders(string x; integer m)

$MP\_next[0] = -1$
for $i = 0$ to $m - 1$ **do**
  $j = MP\_next[i]$.
  while $j \geq 0$ **and** $x[i] \neq x[j]$ **do**
    $j = MP\_next[j]$
  end while
  $MP\_next[i + 1] = j + 1$
end for
return MP_next

Fill up the values in the following table of borders of prefixes, for each of the strings.

strings	0	1	2	3	4	5	6	7	8	9	10
ababa											
abcacabb											
abcacababc											
abacabacab											

What is the complexity of the algorithm?

Solution:

strings	0	1	2	3	4	5	6	7	8	9	10
ababa	-1	0	0	1	2	3					
abcacabb	-1	0	0	0	1	0	1	2	0		
abcacababc	-1	0	0	0	1	0	1	2	1	2	3
abacabacab	-1	0	0	1	0	1	2	3	4	5	6

The algorithm has time complexity $\mathcal{O}(m)$. ■

Exercise 2. *Consider the Morris-Pratt search algorithm, that finds a pattern x of length m in a text y of length n. Consider for this, the pattern $x = aba$ and the text $y = abcacacabac$.*

Algorithm 2 MP(string x, y; integer m, n)

$i = 0, j = 0$
while $j < n$ **do**
  while $(i == m)$ **or** $(i \geq 0$ **and** $x[i] \neq y[j])$ **do**
    $i = MP\_next[i]$
  end while
  $i = i + 1$
  $j = j + 1$
  if $i == m$ **then**
    output: x 'occurs in' y 'at position' $j - i$
  end if
end while

a) *Complete the following table concerning the Preprocessing phase.*

strings	0	1	2	3
$x[i]$	a	b	a	
MP_{next}				

b) *Complete the following table concerning the Searching phase.*

j	0	1	2	3	4	5	6	7	8	9	10
i											

c) *What is the actual value that the algorithm will return?*
d) *What is the search phase complexity?*

Solution:
a) Complete the following table concerning the *Preprocessing* phase.

strings	0	1	2	3
$x[i]$	a	b	a	
MP_next	-1	0	0	1

b) Complete the following table concerning the *Searching* phase.

j	0	1	2	3	4	5	6	7	8	9	10
i	0	1	2	0	1	0	1	0	1	2	3

c) The algorithm will return the position 7 in *abcacacabac*?
d) The *Searching* phase of the algorithm has complexity $\mathcal{O}(n)$. ■

Exercise 3. *Consider the following algorithm calculating the length of the longest borders of all prefixes of a string x of length m, followed by a character different from the one following the prefix, and -1 otherwise.*

Algorithm 3 Compute KMP_next(string x; integer m)

$k = 0$
$j = KMP\_next[0] = -1$
for $i = 0$ to $m - 1$ **do**
  if $x[i] == x[k]$ **then**
    $KMP\_next[i] = KMP\_next[k]$
  else
    $KMP\_next[i] = k$
    do $k = KMP\_next[k]$
    while $k \geq 0$ **and** $x[i] \neq= x[k]$
  end if
  $k = k + 1$
end for
$KMP\_next[m] = k$
return KMP_next

Fill up the values in the following KMP_next table, for each string.

strings	0	1	2	3	4	5	6	7	8	9	10
ababa											
abcacabb											
abcacababc											
abacabacab											

What is the complexity of the algorithm?

Solution:

strings	0	1	2	3	4	5	6	7	8	9	10
ababa	-1	0	-1	0	-1	3					
abcacabb	-1	0	0	-1	1	-1	0	2	0		
abcacababc	-1	0	0	-1	1	-1	0	2	0	0	3
abacabacab	-1	0	-1	1	-1	0	-1	1	-1	0	6

The algorithm has time complexity $\mathcal{O}(m)$. ■