Universitetet i Oslo
Institutt for Informatikk

I. Yu, D. Karabeg

INF2220: algorithms and data structures
Series 10

Topic Text algorithms (Exercises with hints for solution)

Issued: 26. 10. 2016

Exercise 1 Use a *brute force* algorithm to search for the string pattern ("needle") `BAOBAB` in the text

`BESS_KNEW_ABOUT_BAO_AND_BAOBABS`

Solution: [of Exercise 1] You have to check, at all positions in the text $n - m$ (m is the number of characters in the pattern you want to search in a piece of text which has n characters), whether an occurrence of the pattern starts there or not. Then, after each attempt, it shifts the pattern by exactly one position to the right.

Exercise 2 (Bad character shift) Use *bad character shift* to search for the pattern

1. `BAOBAB` in the text `BESS_KNEW_ABOUT_BAOBABS`.
2. `TCCTATTCTT` in the text `TTATAGATCTCGTATTCTTTTATAGATCTCCTATTCTT`.

Solution: [of Exercise 2 (bad character shift)] In both cases the "haystack" is written before the line, and the lines below the line shows the way the "needle" is moved along the "haystack", so in particular it shows the "jumps" caused be the bad character shift.

1.

```
B E S S _ K N E W _ A B O U T _ B A O B A B S
B A O B A B
        B A O B A B
          B A O B A B
                    B A O B A B
                      B A O B A B
```

2.

```
T T A T A G A T C T C G T A T T C T T ...
T C C T A T T C T T
  T C C T A T T C T T
    T C C T A T T C T T
      T C C T A T T C T T
        T C C T A T T C T T
```

```
     ...  T  T  A  T  A  G  A  T  C  T  C  C  T  A  T  T  C  T  T
... C  T  T
   ... C  T  T
     ...  C  T  T
     ...  C  T  A  T  T  C  T  T
     ...  C  C  T  A  T  T  C  T  T
             T  C  C  T  A  T  T  C  T  T
                   T  C  C  T  A  T  T  C  T  T
                      T  C  C  T  A  T  T  C  T  T
                                    T  C  C  T  A  T  T  C  T  T
```

Exercise 3 (Good suffix shift)

1. Construct the *good suffix table* for the pattern `TCCTATTCTT`.
2. Use *good suffix shift* to search for the above pattern in the text

   `TTATAGATCTCGTATTCTTTTATAGATCTCCTATTCTT`.

Solution: [of Exercise 3 (GSS)]

1. Construct the *good suffix table* for the pattern `TCCTATTCTT`.

index	**Mismatch**	**Shift**	`goodCharShift`
0	T	2	goodCharShift[0]==2
1	TT	1	goodCharShift[1]==1
2	CTT	3	goodCharShift[2]==3
3	TCTT	9	goodCharShift[3]==9
4	TTCTT	9	goodCharShift[4]==9
5	ATTCTT	9	goodCharShift[5]==9
6	TATTCTT	9	goodCharShift[6]==9
7	CTATTCTT	9	goodCharShift[7]==9
8	CCTATTCTT	9	goodCharShift[8]==9
9	TCCTATTCTT	9	goodCharShift[9]==9

2.

```
T  T  A  T  A  G  A  T  C  T  C  G  T  A  T  T  C  T  T  ...(cont'd)
T  C  C  T  A  T  T  C  T  T
   T  C  C  T  A  T  T  C  T  T
         T  C  C  T  A  T  T  C  T  T
            T  C  C  T  A  T  T  C  T  T
                        T  C  C  T  A  T  T  C  T  T
```

```
...  T  T  A  T  A  G  A  T  C  T  C  C  T  A  T  T  C  T  T
...  C  C  T  A  T  T  C  T  T
        T  C  C  T  A  T  T  C  T  T
           T  C  C  T  A  T  T  C  T  T
              T  C  C  T  A  T  T  C  T  T
                          T  C  C  T  A  T  T  C  T  T
```

Exercise 4 (Bad character shift & brute force) Is it possible that using *bad character shift* makes more character comparisons than the *brute force* algorithm would make in searching for the same pattern in the same text?

Solution: [of Exercise 4] Yes, e.g., for the pattern `10...0` (the length of the pattern is $m-1$) and the text `0...0` (the length of the text is n) and $n > m$, then
Number of comparisons for *brute force* $= n - m + 1$
Number of comparisons for *bad character shift* $= m(n - m + 1)$

Exercise 5 (Multiple match) Suppose that one or more matches of a pattern exist in a piece of text. By using *bad character shift*, after one match of the pattern is found, how large should a shift be made to search for a next possible match?

Solution: [of Exercise 5] We can shift the pattern exactly in the same manner as we would in the case of a mismatch.

Lab

Exercise 6 Design a *brute force* substring search algorithm that scans the pattern from *right to left.*