**Year 2**

## Data Structures and Algorithms – IT2070

**BSc (Hons) in Information Technology**



**Tutorial 9 – String Matching**

# Semester 2, 2022

Question1

1. If modulo value is *q* = 100, how many spurious hits and valid hits do the **Rabin -Karp matcher** encounter in the text T = 200334002000400 when looking for pattern *P* = 400?
   1. How do you reduce the number of spurious hits in (1)?
2. Draw the statetransitiondiagram for a string-matching automation for the pattern P = ***abaab*** and take the input alphabet as {*a,b*}

1. Following is the **Naïve-String-Matcher** algorithm, which is used to find the occurrence(s) of a pattern string within another string or body of text.

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

***Naïve-String-Matcher*** (T, P)

1. n ← length[T]

2. m ← length[P]

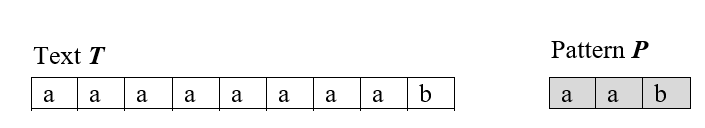
3. for s ← 0 to n-m

4. do if P[1..m] = T[s+1..s+m]

5. then print "Pattern occurs with shift" s

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Given the text and pattern as follows:



Pattern ***P***

1. How many comparisons would occur in this algorithm?
2. Show that worst-case time complexity of the above algorithm is  where *n* is the number of characters in the text and *m* is the number of characters in the pattern.