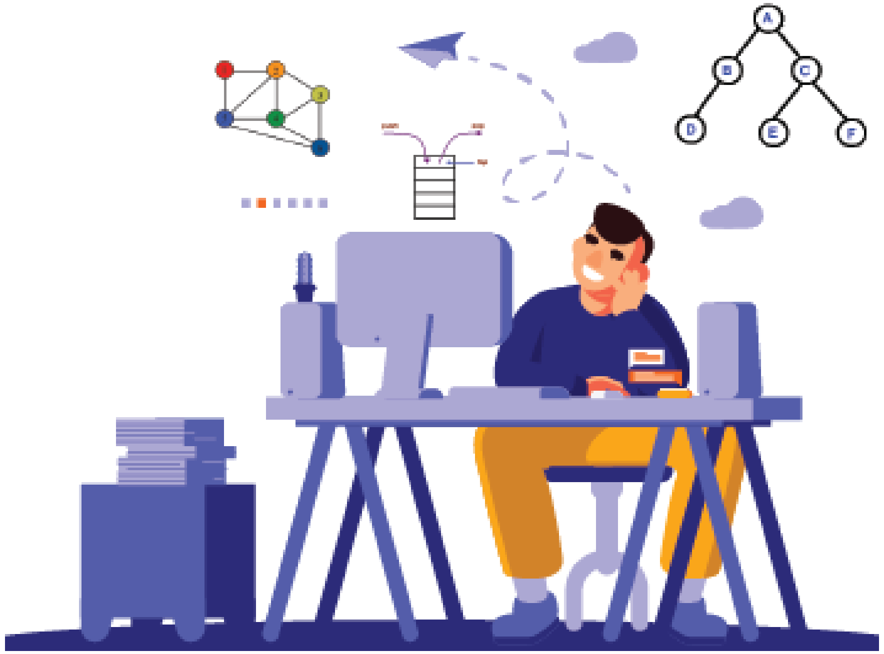
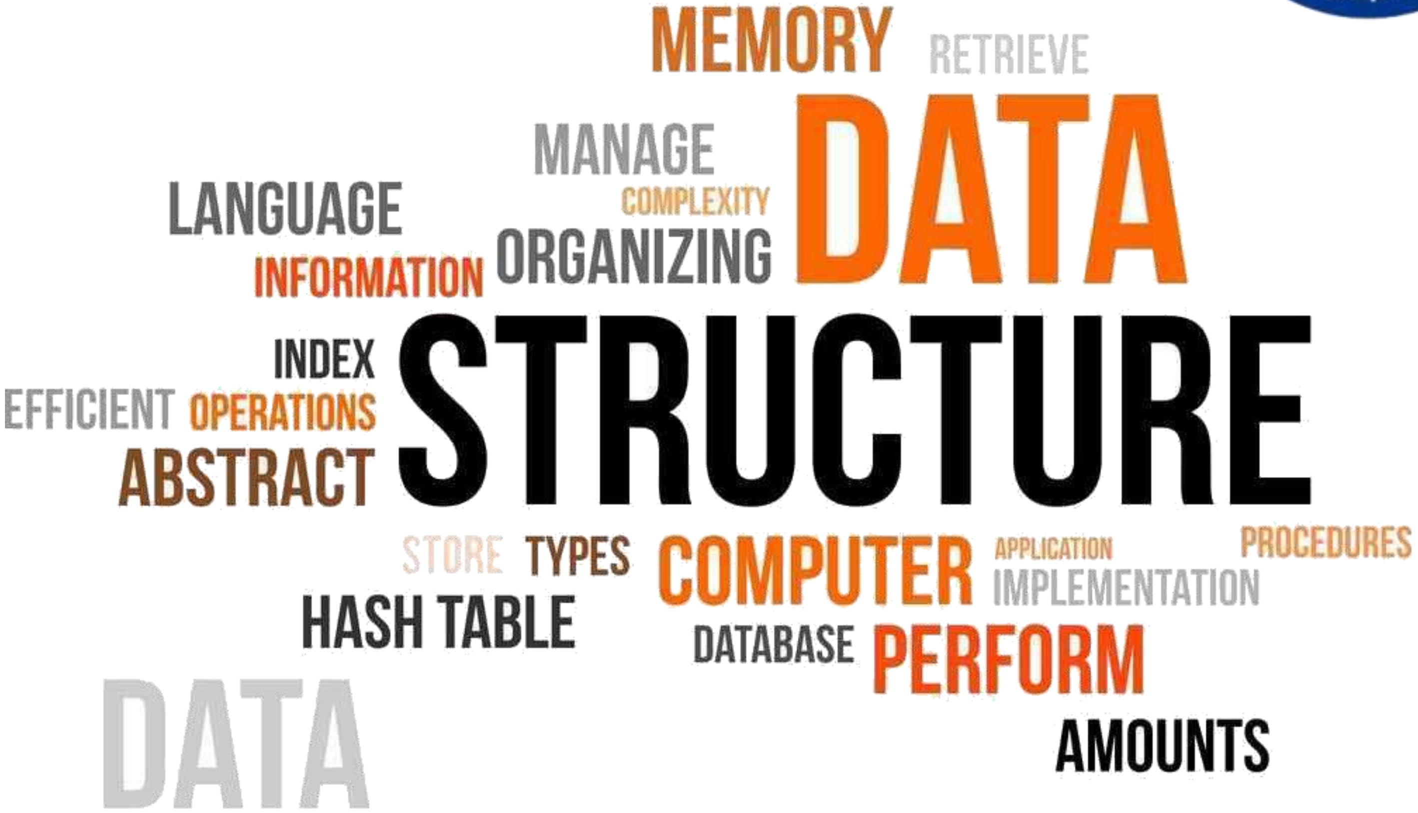


Data Structures

Course code: IT623



Dr. Rahul Mishra
Assistant Professor
DA-IICT, Gandhinagar

Lectures 13

String Processing…

Second Pattern Matching Algorithm :-

* The second pattern matching algorithm uses a table which is derived from a particular pattern P but is independent of the text T. For definiteness, suppose

"$\boxed{P = aabaa}$" pattern

* First we give the reason for the table entries and how they are used.

* Let $T = T_1 T_2 T_3 \ldots$, where T_i denotes the i^{th} character of T; and suppose the first two characters of T match those of P; i.e, suppose $T = aa\ldots$ Then T has one of the following three forms

$\boxed{(i)\ T = aab\ldots,}$ $\boxed{(ii)\ T = aaa\ldots,}$ $\boxed{(iii)\ T = aax}$

Where x is any character different from a or b.

* Let we read T_3 and find that $T_3 = b$. Then we next read T_4 to see if $T_4 = a$,
I → which will give a match of P with W_1.

* Let $\boxed{T_3 = a}$ then we know that $P \neq W_1$; but we also know that $\boxed{W_2 = aa}\ldots$, i.e., that
II → the first two characters of the substring W_2 match those of P. We next read T_4 to see if $\boxed{T_4 = b}$.

*

III → Let $T_3 = x$ then $P \neq W_1$ but we also know that $P \neq W_2$ and $P \neq W_3$. Since x does not appear in P. We next read T_4 to see if $T_4 = a$ i.e, to see if the first character of W_4 matches the first character of P.

Important: >... Conclusion from above three cases:

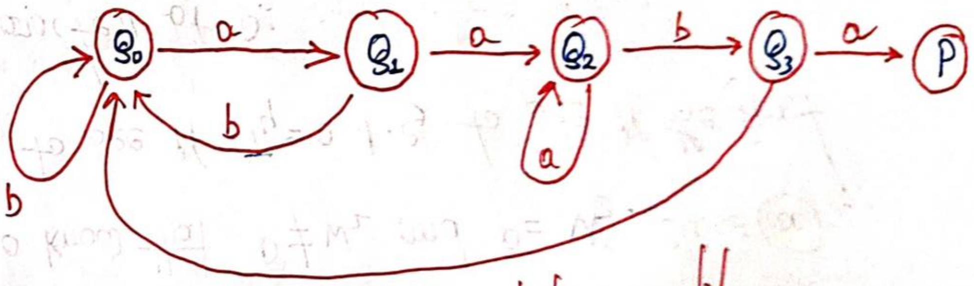
ⓐ When we read T_3 we need only compare T_3 with those character which appear in P. If none of these match, we are in the last case of a character x which does not appear in P

(b) After reading and checking T_3, we next read T_4; we do not have to go back again in the text T.

Consider an example ÷

	a	b	x
Q_0	Q_1	Q_0	Q_0
Q_1	Q_2	Q_0	Q_0
Q_2	Q_2	Q_3	Q_0
Q_3	P	Q_0	Q_0

* Pattern matching table



* Pattern Matching graph

The Figure contains the table that is used in our second pattern matching algorithm for the pattern $\boxed{P = aaba}$.

We alway use italic letters to represent pattern in a string.

* The table is obtained as follows.

* Let Q_i denote the initial substring of P of length i; hence

$Q_0 = \Lambda$	, $Q_1 = a$	, $Q_2 = a^2$	, $Q_3 = a^2b$	, $Q_4 = a^2ba = P$

$Q_0 = \Lambda$ is the empty string.

> The rows of the table are labeled by these initial substring of P excluding P itself.

> The columns of the table are labeled a, b, and x, where x represents any character that doesn't appear in the pattern P.

* Let f be the function determined by the table; i.e, let denote the entry in row Q_i and column t.

The entry $f(q_i, t)$ is defined to be the longest q that appears as a terminal substring in the string $q_i t$.

— a^2 is the longest q that is a terminal substring of $q_2 a = a^3$, so $\boxed{f(q_2, a) = q_2}$

— Λ is the longest q that is a terminal substring of $q_1 b = ab$, so $\boxed{f(q_1, b) = q_0}$

— a is the longest q that is a terminal substring of $q_0 a = a$, so $\boxed{f(q_0, a) = q_1}$

— Λ is the longest q that is a terminal substring of $q_3 a = a^3 bx$, so $\boxed{f(q_3, x) = q_0}$

* $q_1 = a$ is a terminal substring of $\boxed{q_2 a = a^3}$ we have $f(q_2, a) = q_2$ because q_2 is also a terminal substring $\boxed{q_2 a = a^3}$.

→ Pattern $P = aaba$, . Let $T = T_1 T_2 T_3 \ldots T_N$ denote that n character-string text which is searched for the pattern P.

→ Begining with the initial state S_0 and using the text T, will be obtain a sequence of states $S_1, S_2, S_3 \ldots$ as follows.

→ We let $S_1 = S_0$ and we read the first character T_1. From either the table on the graph in Figure. ---(S_1, T_1) yield new state. and so on.

Notably: * a) Some state $S_K = P$, the desired pattern. In this case, P does not appear in T and its index is K − LENGTH(P). Example: aabcaba

b) No state $S_1, S_2, \ldots, S_{N+1}$ is equal to P. In this case, P does not appear in T. Example: abcaabaca

(Pattern Matching). The pattern matching table $F(Q_1, T)$ of a pattern P is in memory, and the input is an N-character string $T = T_1T_2 \ldots T_N$. This algorithm finds the INDEX of P in T.

1. [Initialize.] Set $K := 1$ and $S_1 = Q_0$
2. Repeat Steps 3 to 5 while $S_K \neq P$ and $K \leq N$.
3. Read T_K.
4. Set $S_{K+1} := F(S_K, T_K)$. [Finds next state.]
5. Set $K := K + 1$. [Updates counter.]

   [End of Step 2 loop.]
6. [Successful?]

   If $S_K = P$, then:

   INDEX = K – LENGTH(P).

   Else:

   INDEX = 0.

   [End of If structure.]
7. Exit

Pattern Matching Second Algorithm

* The running time of the above algorithm is proportional to the number of times the step 2 loop is executed.

* The worst case occurs when all of the text **T** is read, i.e., when the loop is executed $n =$ LENGTH(T) times.

* Thus, we can conclude that the complexity of this pattern matching algorithm is $\boxed{O(n)}$.

Two numerical examples –

ⓐ

ⓑ

Consider the pattern $P = aaabb$. Construct the table and the corresponding labeled directed graph used in the "fast," or second pattern matching, algorithm.

First list the initial segments of P:

$$Q_0 = \Lambda, \quad Q_1 = a, \quad Q_2 = a^2, \quad Q_3 = a^3, \quad Q_4 = a^3b, \quad Q_5 = a^3b^2$$

For each character t, the entry $f(Q_i, t)$ in the table is the largest Q which appears as a terminal substring in the string Q_it. We compute:

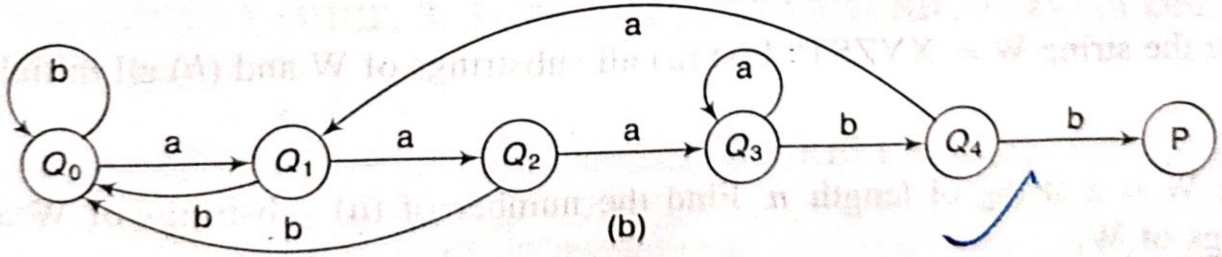
$$\begin{array}{lllll} f(\Lambda, a) = a, & f(a, a) = a^2, & f(a^2, a) = a^3, & f(a^3, a) = a^3, & f(a^3b, a) = a \\ f(\Lambda, b) = \Lambda, & f(a, b) = \Lambda, & f(a^2, b) = \Lambda, & f(a^3, b) = a^3b, & f(a^3b, b) = P \end{array}$$

Example 1

Hence the required table appears in Fig. 3.10(a). The corresponding graph appears in Fig. 3.10(b), where there is a node corresponding to each Q and an arrow from Q_i to Q_j labeled by the character t for each entry $f(Q_i, t) = Q_j$ in the table.

	a	b
Q_0	Q_1	Q_0
Q_1	Q_2	Q_0
Q_2	Q_3	Q_0
Q_3	Q_3	Q_4
Q_4	Q_1	P

(a)



(b)

Find the table and corresponding graph for the second pattern matching algorithm where the pattern is $P = ababab$.

The initial substrings of P are:

$$Q_0 = \Lambda, \quad Q_1 = a, \quad Q_2 = ab, \quad Q_3 = aba, \quad Q_4 = abab, \quad Q_5 = ababa, \quad Q_6 = ababab = P$$

The function f giving the entries in the table follows:

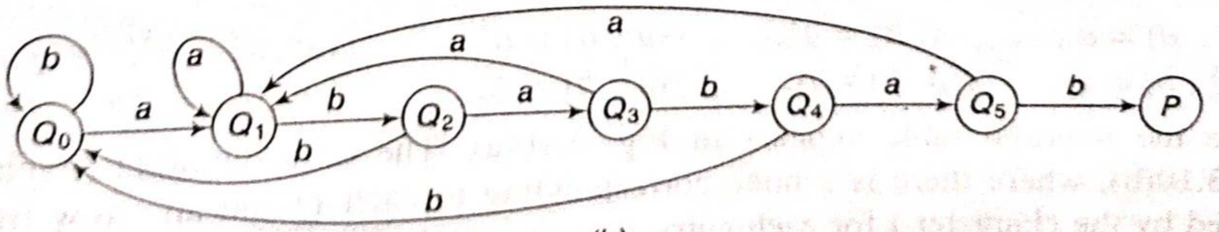
$f(\Lambda, a) = a$	$f(\Lambda, b) = \Lambda$
$f(a, a) = a$	$f(a, b) = ab$
$f(ab, a) = aba$	$f(ab, b) = \Lambda$
$f(aba, a) = a$	$f(aba, b) = abab$
$f(abab, a) = ababa$	$f(abab, b) = \Lambda$
$f(ababa, a) = a$	$f(ababa, b) = P$

Example 2

The table appears in Fig. 3.11(a) and the corresponding graph appears in Fig. 3.11(b).

	a	b
Q_0	Q_1	Q_0
Q_1	Q_1	Q_2
Q_2	Q_3	Q_0
Q_3	Q_1	Q_4
Q_4	Q_5	Q_0
Q_5	Q_1	P

(a)



(b)