**KMP (Knuth-Morris-Pratt) algorithm**

Knuth Morris Pratt (KMP) is an algorithm, which checks the characters from left to right. When a pattern has a sub-pattern appears more than one in the sub-pattern, it uses that property to improve the time complexity, also for in the worst case.

The time complexity of KMP is O(n).

Knuth-Morris and Pratt introduce a linear time algorithm for the string matching problem. A matching time of O (n) is achieved by avoiding comparison with an element of 'S' that have previously been involved in comparison with some element of the pattern 'p' to be matched. i.e., backtracking on the string 'S' never occurs

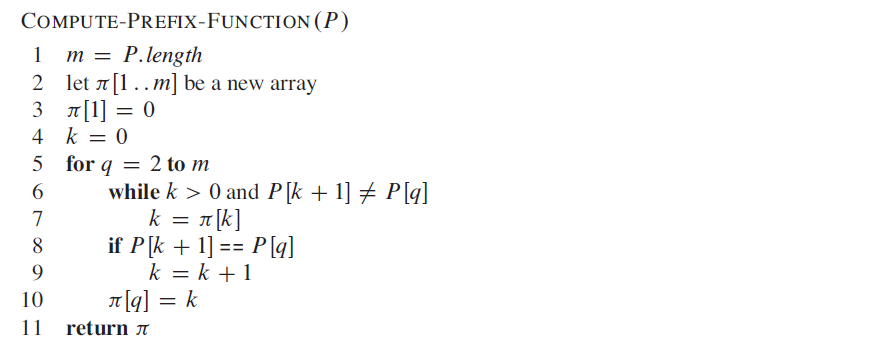
Components of KMP Algorithm:

**1. The Prefix Function (Π):** The Prefix Function, Π for a pattern encapsulates knowledge about how the pattern matches against the shift of itself. This information can be used to avoid a useless shift of the pattern 'p.' In other words, this enables avoiding backtracking of the string 'S.'

**2. The KMP Matcher:** With string 'S,' pattern 'p' and prefix function 'Π' as inputs, find the occurrence of 'p' in 'S' and returns the number of shifts of 'p' after which occurrences are found.

## The Prefix Function(Π)

Following pseudo code compute the prefix function Π

****

## Running Time Analysis:

In the above pseudo code for calculating the prefix function, the for loop from step 4 to step 10 runs 'm' times. Step1 to Step3 take constant time. Hence the running time of computing prefix function is O (m).

**Example:** Compute Π for the pattern 'p' below:

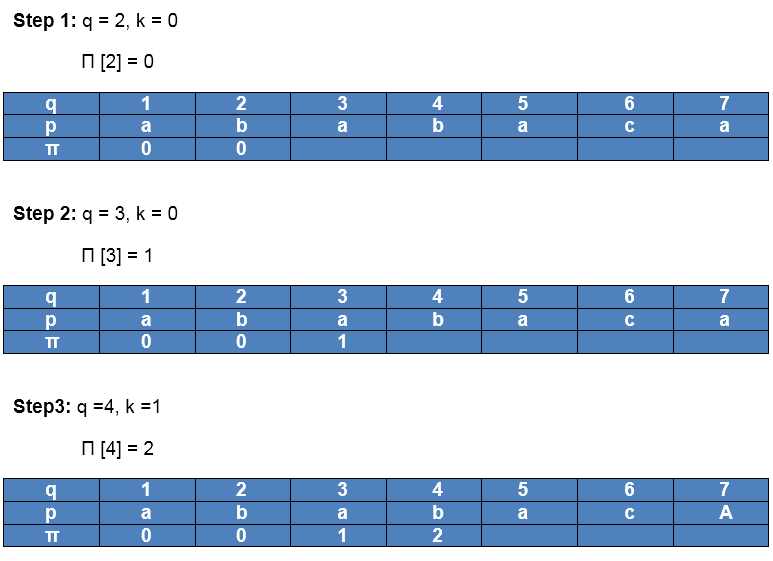
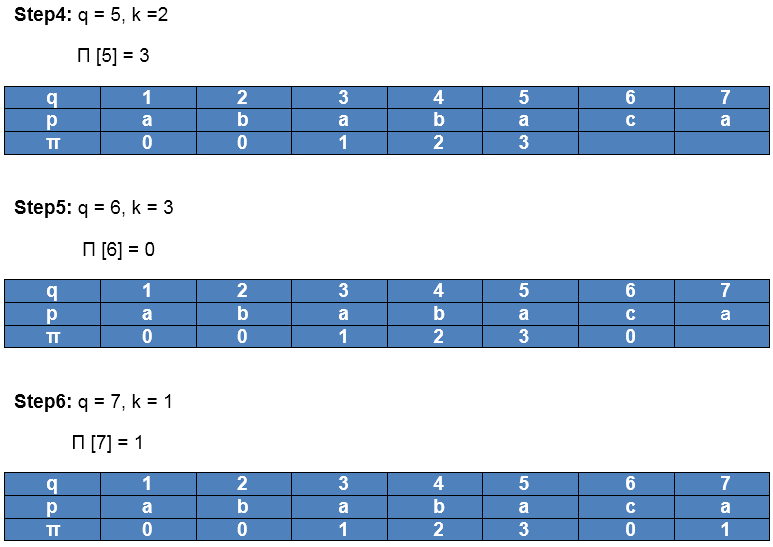
Knuth-Morris-Pratt Algorithm

**Solution:**

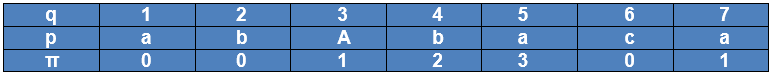
Initially: m = length [p] = 7

Π [1] = 0

k = 0

After iteration 6 times, the prefix function computation is complete:



## The KMP Matcher:

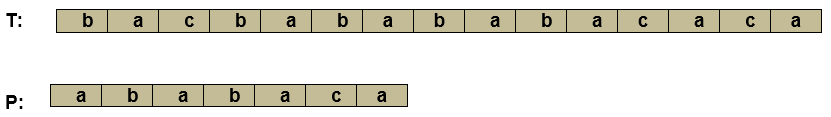
The KMP Matcher with the pattern 'p,' the string 'S' and prefix function 'Π' as input, finds a match of p in S. Following pseudo code compute the matching component of KMP algorithm:

## 

## Running Time Analysis:

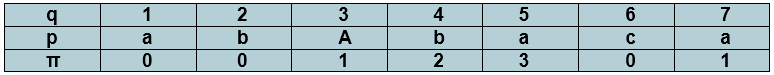
The for loop beginning in step 5 runs 'n' times, i.e., as long as the length of the string 'S.' Since step 1 to step 4 take constant times, the running time is dominated by this for the loop. Thus running time of the matching function is O (n).

**Example:** Given a string 'T' and pattern 'P' as follows:



Let us execute the KMP Algorithm to find whether 'P' occurs in 'T.'

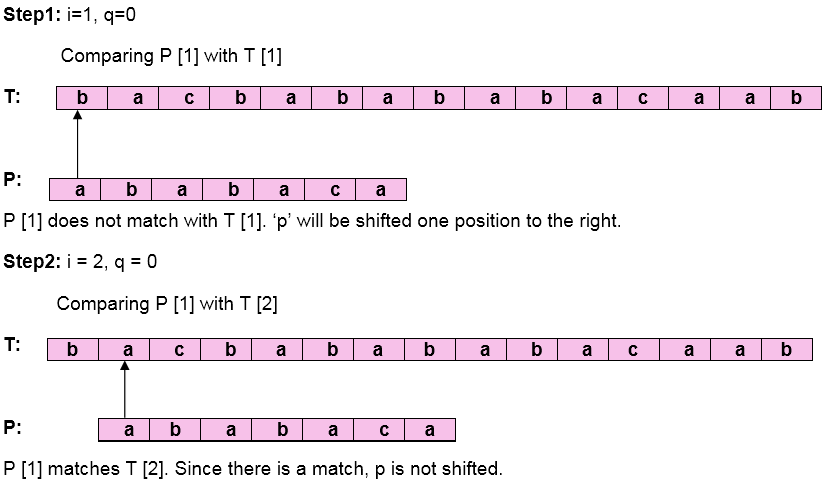
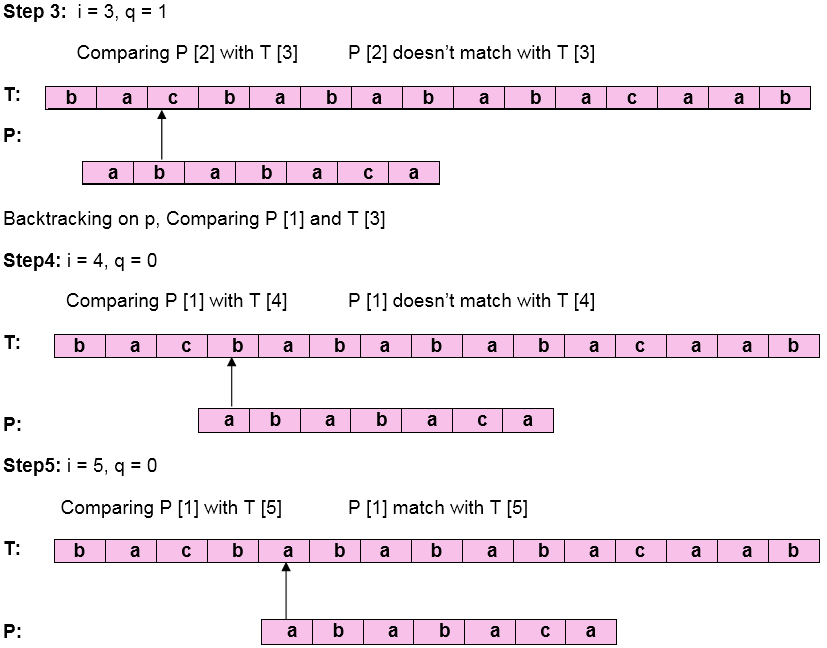
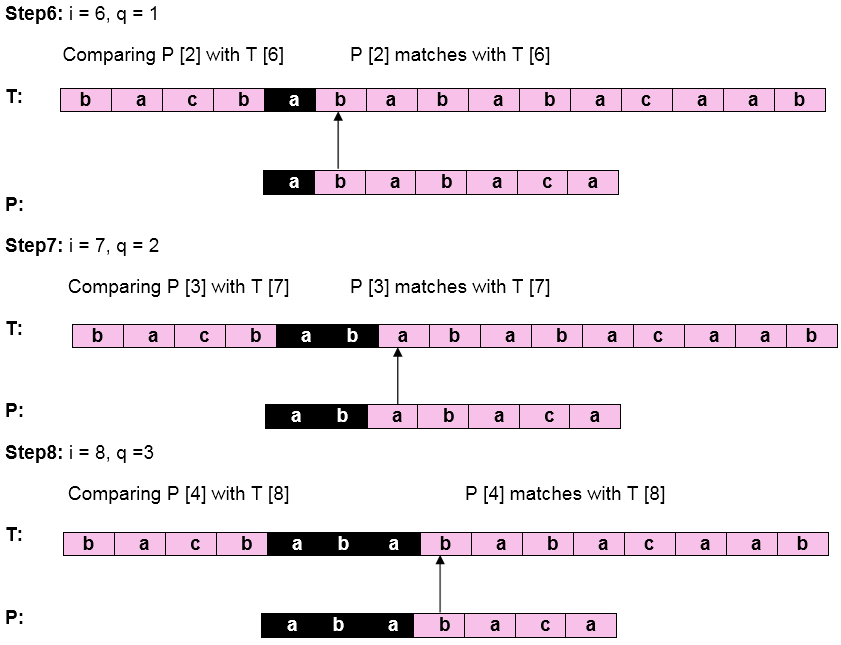
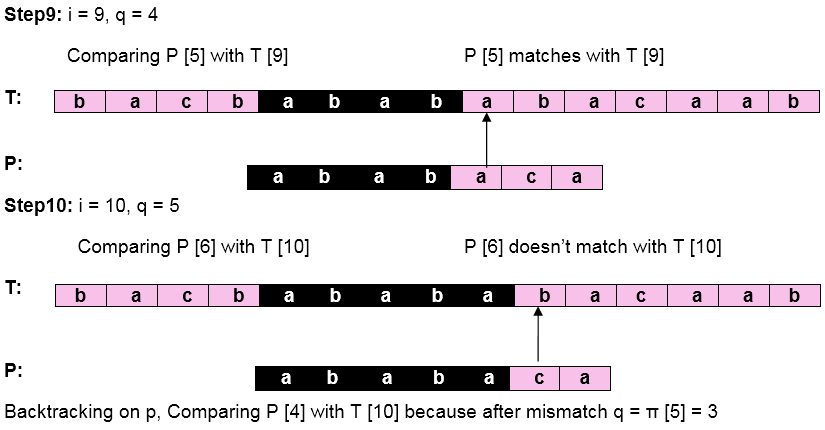
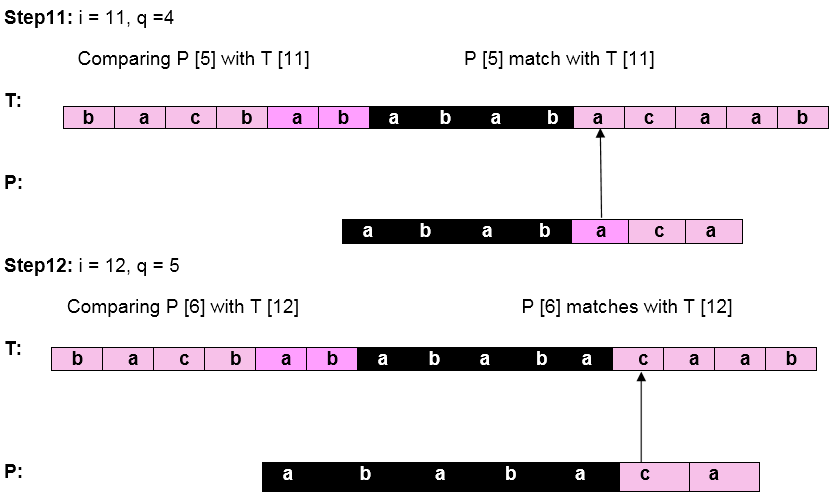
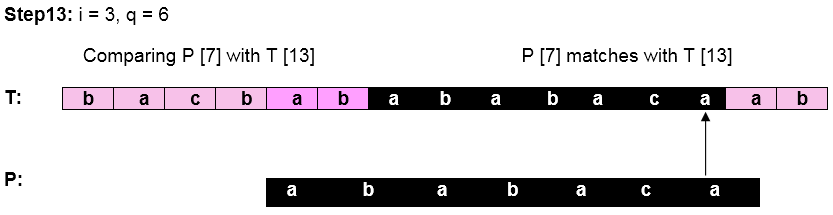
For 'p' the prefix function, ? was computed previously and is as follows:



**Solution:**

Initially: n = size of T = 15

m = size of P = 7

Pattern 'P' has been found to complexity occur in a string 'T.' The total number of shifts that took place for the match to be found is i-m = 13 - 7 = 6 shifts.

**RELEVANT READING MATERIAL AND REFERENCES:**

**Source Notes:**

1. <https://www.javatpoint.com/daa-knuth-morris-pratt-algorithm>
2. <https://www.tutorialspoint.com/Knuth-Morris-Pratt-Algorithm>

**Lecture Video:**

1. https://youtu.be/V5-7GzOfADQ

**Online Notes:**

1. <http://vssut.ac.in/lecture_notes/lecture1428551222.pdf>

**Text Book Reading:**

1. Cormen, Leiserson, Rivest, Stein, “*Introduction to Algorithms*”, Prentice Hall of India, 3rd edition 2012. problem, Graph coloring.

**In addition: PPT can be also be given.**