String Matching Algorithm

- String matching algorithms are normally used in text processing
- String matching means finding one or more occurrences of string in the text.

- Applications :-

  - String matching algorithms search for perticular patterns in DNA sequence.

- Internet search engines also use it to find web pages relevant to queries.

string matching problem is defined as follows

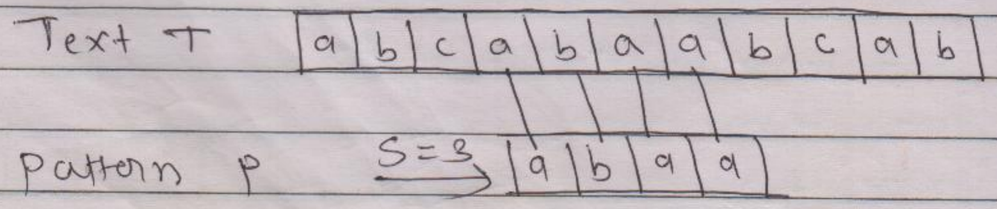
The text is an array $T[1\cdots n]$ of length n. and the pattern is an array $P[1\cdots m]$ of length $m \leq n$

The elements of P and T are characters drawn from finite alphabet Σ.

eg: $\Sigma = \{0,1\}$ or $\Sigma = \{a,b,\cdots z\}$

The character arrays T and P are often called strings of characters.

The string matching problem is the problem of finding all valid shifts with which a given pattern P occurs in a given text T.



Naive String Matching Algorithm :-

This is the simplest method which works using Brute Force approach.

- This algorithm performs a checking at all positions in the text between 0 to n-m whether an occurrence of the pattern starts there or not.

- Then it shifts patterns by exactly one position to the right.

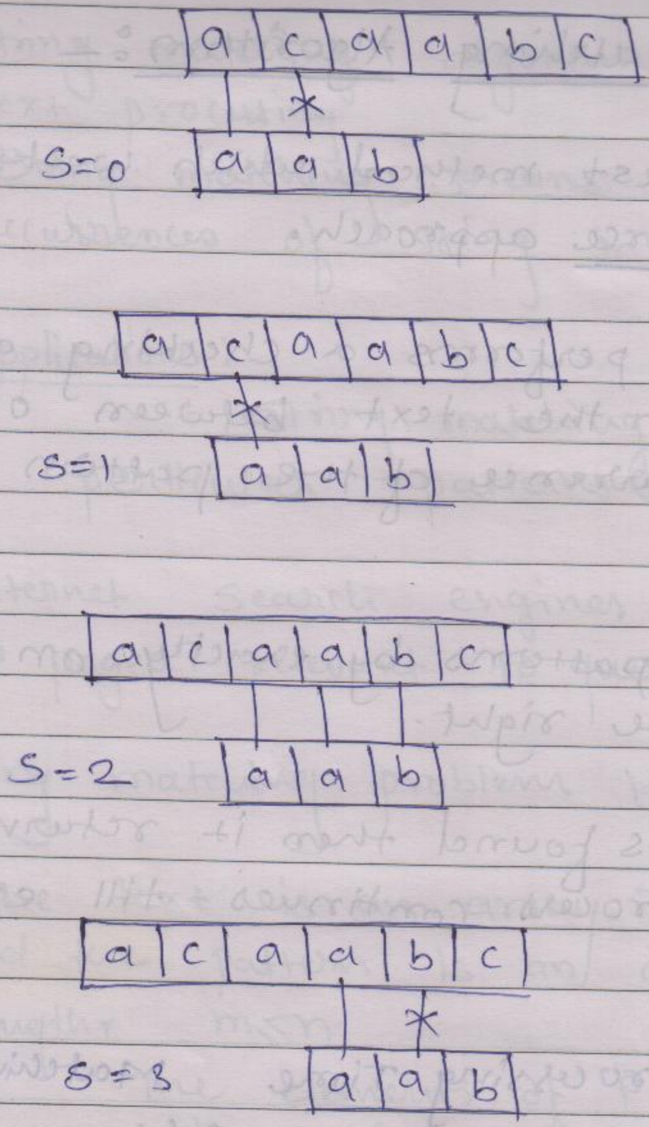
- If the match is found then it returns shift value and process continues till end of the text.

Algorithm	preprocessing Time	Matching Time
Naive	0	$O((n-m+1)m)$

Algorithm :-

```
Naive - String - Matcher (T, P)
  n = T. length
  m = P. length
  for S = 0 to n-m
     if P[1...m] == T[S+1 .... S+m]
        print " pattern occurs with shift " s
```

eg :


a c a a b c
s=0
a a b
a c a a b c
s=1
a a b
a c a a b c
s=2
a a b
a c a a b c
s=3
a a b