Data Structures and Algorithms

Prof. Ganesh Ramakrishnan,
Prof. Ajit Diwan,
Prof. D.B. Phatak

Department of Computer Science and Engineering
IIT Bombay

Session: String Matching Algorithms

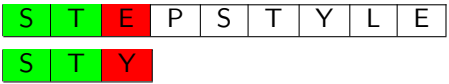
String Matching Problem[1]

- Finding all valid shifts with which a given pattern P occurs in a given text T

[1] Chapter 32, CLRS, Third Edition

Illustration

$s = 0$



Illustration

$s = 0$

S	T	E	P	S	T	Y	L	E

S	T	Y

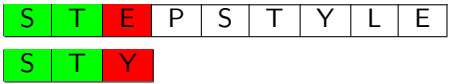
$s = 1$

S	T	E	P	S	T	Y	L	E

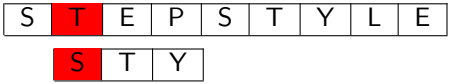
S	T	Y

Illustration

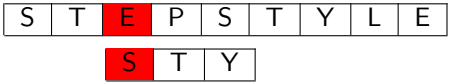
$s = 0$



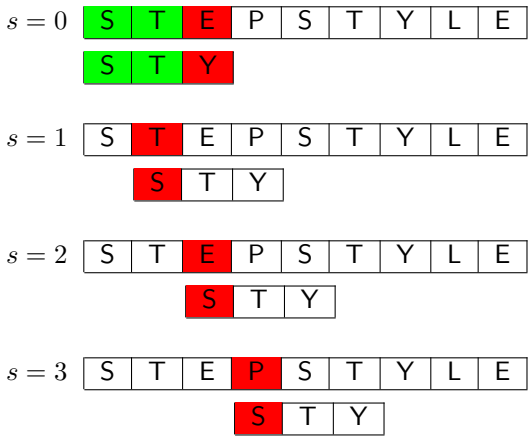
$s = 1$



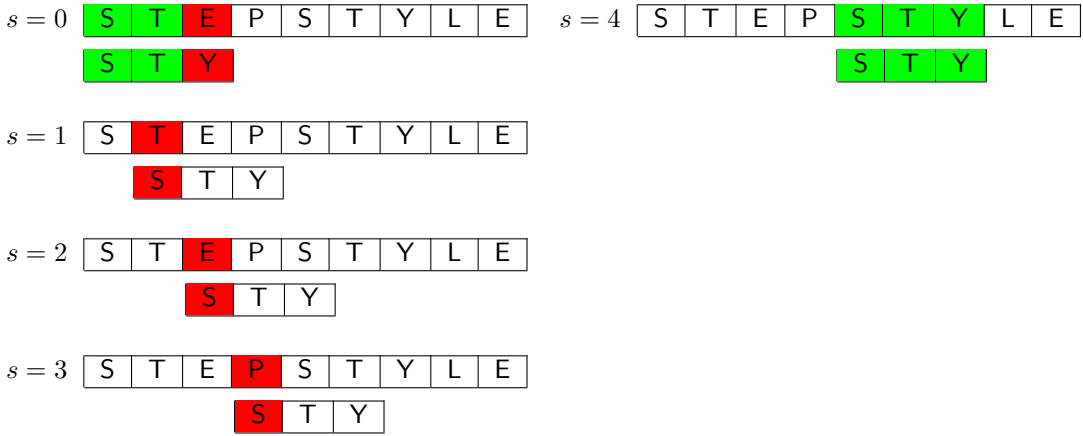
$s = 2$



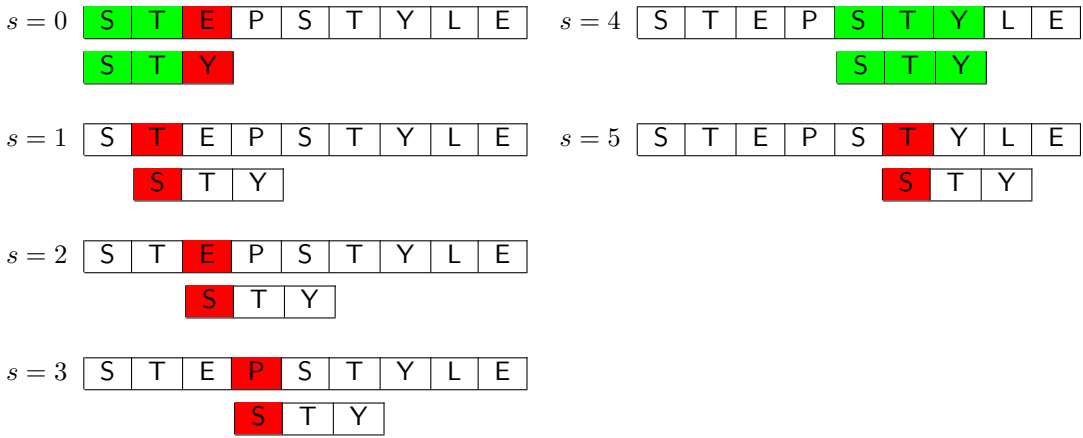
Illustration



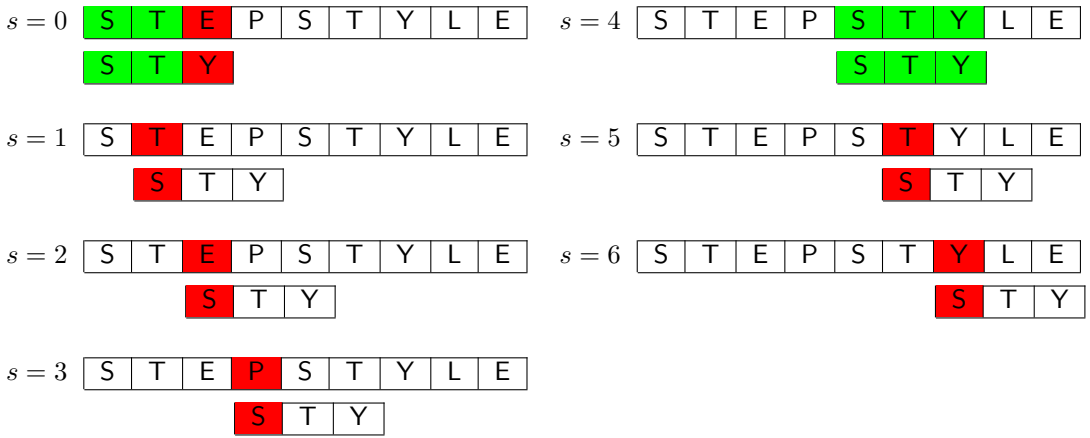
Illustration



Illustration



Illustration



Naive String Matching Algorithm

Algorithm NaiveStringMatchingAlgorithm(T, P)
Input Text T of size n and Pattern P of size m
Define s as the shift index to T
for $s \in (0...n-m)$ **do**
    $j = 0$
    while $j < m$ & $T[s+j] = P[j]$ **do**
        $j = j + 1$
    end while
    if $j = m$ **then**
        print 'Valid at shift s'
    end if
end for

Figure: Naive String Algorithm

Analysis of Naive String Matching Algorithm

Algorithm NaiveStringMatchingAlgorithm(T, P)
Input Text T of size n and Pattern P of size m
Define s as the shift index to T
for $s \in (0...n-m)$ **do**
  $j = 0$
  while $j < m$ & $T[s+j] = P[j]$ **do**
    $j = j + 1$
  end while $\Longrightarrow$ c_1 $\times$ m times
  if $j = m$ **then**
    print 'Valid at shift s'
  end if
end for $\Longrightarrow$ c_2 $\times$ $(n-m+1)$ times

Figure: Naive String Matching Algorithm

$T(n) = c_1c_2(n-m+1)m = O((n-m+1)m)$

**Note: Preprocessing time is 0

Thank you