

Randomized String Matching

procedure **Rabin-Karp-Matcher**(T, P, d, q):

{ Input: Text string T , pattern P , $d = |\Sigma|$, prime q }

{ Output: Locations of P in T }

$n = T.length$

$m = P.length$

$p = 0; t_0 = 0$

for $i = 1$ **to** m

$p = (dp + P[i]) \bmod q$

$t_0 = (dt_0 + T[i]) \bmod q$

for $s = 0$ **to** $n - m$

if $p == t_s$

if $P[1..m] == T[s + 1 \dots s + m]$

print "Pattern occurs with shift" s

if $s < n - m$

$t_{s+1} = (d(t_s - d^{m-1} T[s + 1]) + T[s + m + 1]) \bmod q$

Finite Automata

```
procedure Matcher( $T, \delta, m$ ):  
{ Input: Text string  $T$ , transition function  $\delta$  for pattern  $P$ ,  $m = |P|$  }  
{ Output: Locations of  $P$  in  $T$  }  
   $n = T.length$   
   $q = 0$   
  for  $i = 1$  to  $n$   
     $q = \delta(q, T[i])$   
    if  $q == m$   
       $s = i - m$   
      print "Pattern occurs with shift"  $s$ 
```

KMP Matcher Algorithm

```
procedure KMP-Matcher( $T, P$ ):  
{ Input: Text string  $T$ , pattern  $P$  }  
{ Output: Locations of  $P$  in  $T$  }  
   $n = T.length$   
   $m = P.length$   
   $\pi = \text{Compute-Prefix-Function}(P)$   
   $q = 0$   
  for  $i = 1$  to  $n$   
    while  $q > 0$  and  $P[q + 1] \neq T[i]$   
       $q = \pi[q]$   
    if  $P[q + 1] == T[i]$   
       $q = q + 1$   
    if  $q == m$   
      print "Pattern occurs with shift"  $s$   
       $q = \pi[q]$ 
```

KMP Algorithm to Compute Prefix Function

procedure **Compute-Prefix-Function**(P):

{ Input: Pattern P }

{ Output: Prefix function π for P }

$m = P.length$

$\pi[1] = 0$

$k = 0$

for $q = 2$ **to** m

while $k > 0$ and $P[k + 1] \neq P[q]$

$k = \pi[k]$

if $P[k + 1] == P[q]$

$k = k + 1$

$\pi[q] = k$