Algorithms Design and Analysis [ETCS-301]

Dr. A K Yadav
Amity School of Engineering and Technology
(affiliated to GGSIPU, Delhi)
akyadav1@amity.edu
akyadav@akyadav.in
www.akyadav.in
+91 9911375598

October 31, 2019

String Matching I

- $\sum$ is a set of characters.
- $T = T[1..n]$ and $P = P[1..m]$ is a string of characters and each character belongs to $\sum$ that is $T[i] \in \sum$ for $1 \leq i \leq n$ and $P[i] \in \sum$ for $1 \leq i \leq m$.
- T is a text of length n from which we want to find out the occurrences of the pattern P of length m.
- The length of the pattern may be less then or equal to length of the text i.e. $m \leq n$
- The pattern P may be any where in the text T i.e. $T[s+1..s+m] = P[1..m]$ or $T[s+j] = P[j]$ for $1 \leq j \leq m$
- s is know as shift and value of s may be from 0 to $n-m$; $n-m+1$ total values.
- When $s = 0$ the first m characters of text $T[1..m]$ will be checked with P.

String Matching II

- That is if s is the shift then the position of the text to be matched is $s+1$
- Pattern P occurs in text T with shift s or beginning at position $s+1$ if $0 \leq s \leq n-m$ and $T[s+1..s+m] = P[1..m]$ that is if $T[s+j] = P[j]$ for $1 \leq j \leq m$
- If P occurs with shift s in T , then we call s a valid shift; otherwise, we call s an invalid shift.
- So the string-matching problem is the problem of finding all valid shifts s with which a given pattern P occurs in a given text T.

$$s = \begin{cases} valid & \text{if } 0 \leq s \leq n-m \text{ and } T[s+1..s+m] = P[1..m] \\ invalid & \text{if } 0 \leq s \leq n-m \text{ and } T[s+1..s+m] \neq P[1..m] \end{cases}$$

Notation and terminology I

- $\sum^*$ is the set of all finite-length strings formed using characters from the alphabet $\sum$.
- The zero-length empty string, denoted ε is also belongs to $\sum^*$
- The length of a string x is denoted $|x|$.
- The concatenation of two strings x and y, denoted xy, has length $|x| + |y|$ and consists of the characters from x followed by the characters from y.
- String w is a prefix of a string x if $x = wy$ for some string $y \in \sum^*$ and represented as $w \sqsubset x$ and $|w| \leq |x|$
- String w is a suffix of a string x if $x = yw$ for some string $y \in \sum^*$ and represented as $w \sqsupset x$ and $|w| \leq |x|$
- The empty string ε is both a suffix and a prefix of every string x

Notation and terminology II

- For any strings x and y and any character a, we have $x \sqsupset y$ if and only if $xa \sqsupset ya$.
- P_k is the k-character prefix $P[1..k]$ of the pattern $P[1..m]$. $P_0 = \varepsilon, P_m = P = P[1..m]$
- T_k is the k-character prefix $T[1..k]$ of the text $T[1..n]$. $T_0 = \varepsilon, T_n = T = T[1..n]$
- Finding all shifts s in the range $0 \leq s \leq n - m$ such that $P \sqsupset T_{s+m}$ is the string-matching problem.

Overlapping-suffix lemma

Statement: Suppose x, y and z are strings such that $x \sqsupset z$ and $y \sqsupset z$. Then 3 cases will be there:

1. If $|x| < |y|$ then $x \sqsupset y$
2. If $|y| < |x|$ then $y \sqsupset x$
3. If $|x| = |y|$ then $x = y$

Overlapping-prefix lemma

Statement: Suppose x, y and z are strings such that $x \sqsubset z$ and $y \sqsubset z$. Then 3 cases will be there:

1. If $|x| < |y|$ then $x \sqsubset y$
2. If $|y| < |x|$ then $y \sqsubset x$
3. If $|x| = |y|$ then $x = y$

Naive string-matching algorithm I

NAIVE-STRING-MATCHER(T,P)

1. $n = length(T)$
2. $m = length(P)$
3. for $s = 0$ to $n - m$
4. if $P[1..m] = T[s + 1..s + m]$
5. print pattern occurs at position $s + 1$.

Naive string-matching algorithm II

NAIVE-STRING-MATCHER(T,P)

1. $n = length(T)$
2. $m = length(P)$
3. for $s = 0$ to $n - m$
4. $\quad valid = 1$
5. $\quad$ for $j = 1$ to m
6. $\quad\quad$ if $P[j] \neq T[s + j]$
7. $\quad\quad\quad valid = 0$
8. $\quad$ if $valid = 1$
9. $\quad\quad$ print pattern occurs with sift s.

Complexity of the algorithm is $O((n - m + 1)m)$

Thank you

Please send your feedback or any queries to akyadav1@amity.edu, akyadav@akyadav.in or contact me on +91 9911375598