Fundamentals of Algorithms
CS502-Spring 2011

SOLUTION ASSIGNMENT4

Deadline

Your assignment must be uploaded/submitted at or before **30th June 2011.**

Uploading instructions

Please view the **assignment submission process** document provided to you by the Virtual University to upload the assignment.

Rules for Marking

It should be clear that your assignment will not get any credit if:

- The assignment is submitted after due date.
- The submitted assignment does not compile or run.
- **The assignment is copied.**

Objectives

This assignment is especially designed to make your vision clear for the understanding of existing material in form of research and new devised algorithms. This is application level for your course of algorithms analysis .Sole purpose of this assignment is to think critically and understand the logics on your own behalf while reading the search material with peace of mind and with clear perceptions.

Guidelines

You are given a paper to solve this assignment and after reading this paper with concentration you will be able to answer the asked queries.
You are supposed to answer the asked points no further details are required. This is just for your ease and understanding of the material.

What to submit

You are supposed to submit ***only one word file*** after reading the paper given to you; you are bound to answer in the following format.

For your easiness we will explore this paper only for single patterns not for multiple patterns however for your perceptions you can read other material as well .

Your task will be done if you read first three pages of the given paper with concentration. Be precise to get full marks.

Introduction of Paper: 1.5

In under consideration paper the authors discussed the algorithm of multiple string-patterns matching; the multiple string pattern matching problems is to find all occurrences of multiple input patterns, P_1….Pn in text. In this paper authors proposed a new multiple string–pattern matching based on compact encoding and hashing scheme and compared their simple and efficient algorithm by taking sample patterns with grep and agrep to make proof of efficiency in certain cases of their purposed algorithm.

What technique has been discussed in this paper for string pattern matching and its benefit? 1.5

The semi numerical approach/compact encoding scheme in which pattern and text is represented in bits which results in comparing many patterns and text characters simultaneously which can lead to efficient technique for finding multiple patterns concurrently.

Apply discussed encoding function on the following pattern: 5

P=atacg

TS= agctatacgtagac

Note : This is actually very simple initiative to encode the complex patterns and it is actually DNA sequence consisting of four letters i.e. “a”, “t”,”g” and “c” and these are four distinct letters so we can use two bits for these letters to encode our patterns say “a=00”,”t=01”,”g=10” and “c=11” .

Thus above Pattern and text string will become”

P=atacg

P= 00 01 00 11 10

TS= agctatacgtagac

TS=00 10 11 01 00 01 00 11 10 01 00 10 00 11.

Apply the algorithm discussed in paper for finding SINGLE pattern taking the following example 10

Initial thinking points:

If we have P pattern having D distinct characters and "E" be the least integer such that $2^E >= (D+1)$. Then we can encode any symbol in P and T with E bits by assigning distinct E bits for each character in P and assigning distinct E bits for any character that does not occur in P but occurs in T. The following example illustrates this scheme.

P=hashing

S=Algorithm hashing and compact encoding

We have pattern P=hashing and text T=Algorithm hashing and compact encoding . Here in pattern distinct characters are 6 and according to our generic formula above we can encode this pattern in "3" bits as $2^3 >= (6+1)$ and to encode the pattern we need an encode function and the whole procedure will be done in this way.

Encode(h)=001,Encode(a)=010 Encode(s)=011 Encode(i)=100 Encode(n)=101 Encode(g)=110 and for any other character which is not in pattern but occur s in T we can assign Encode(--)=000

Now P will be after encoding 001 010 011 001 100 101 110 and

T= 000 000 110 000 000 100 000 001 000 001 010 011 001 100 101 110 010 101 000 000 000 000 000 010 011 000 000 101 000 000 000 100 101 110.

Here to proceed there is concept of computer word which is 32 bits long in our case; elaboration is done stepwise a:

Finding the single pattern —Shift and OR operations help us to store the pattern in word for example we have pattern ''hashing and we have already encoded it. To store it in computer word initializes the word with all zeros now :

A) Perform logical OR between Encode (h) =001 and computer word W.

B) Shift W left "3" bits and make logical OR again of it with Encode(a) note here we are making shifts according to number of "E" bits which we have discovered before,

C) Perform step B until we get the last character of pattern being processed.

At last the word achieved is: 000 000 000 00 001 010 011 001 100 101 110 here note the lower 21 bits are representation of pattern "hashing"

Here the same process will be required to store the text in T variable. To find the pattern a mask concept has been described say it the PMASK which contains all ones ' for pattern length and zero for others. Here the PMASK for under consideration pattern is **00000000000111111111111111111111**

Note

This PMASK is the crux of logic to determine the pattern is in T or not.

Perform **(T (AND) PMASK) XOR (P)** if the result is zero pattern does occur other wise not. Here we will encode each text word into 32 bits word of computer to perform the operation.

T(AND)PMASK

00000000000110000000100000001000

AND

00000000000111111111111111111111

=00000000000110000000100000001000

XOR

000 000 000 00 001 010 011 001 100 101 111

(T AND PMASK) XOR P=000 000 000 00 11101001110110010 0111

Which is not all zero means there is no occurrence at this level.

Now other case while scanning “g” of hashing we perform the same operation again:

T(AND)PMASK

000 000 000 00 001 010 011 001 100 101 110

AND

00000000000111111111111111111111

=000 000 000 00 001 010 011 001 100 101 110

XOR

000 000 000 00 001 010 011 001 100 101 110

= 000 000 000 000 000 000 000 000 000 000 00

Which means pattern occurs at this stage.

Similarly other text “and compact encoding” will be compared obviously these words again results in non zero means not the occurrence of the pattern.

And we are done.

Conclusions 2

The purposed algorithm is better and provides better results for certain type of patterns for example as of DNA search cases. To make this algorithm more efficient we need to explore adaptive string matching techniques.