UE14CS311 – Advanced Algorithms (5th Sem Elective)

Assignment 2 : String Search and Suffix Trees

1. **Guidelines:**

a) Code can be developed in any of C/C++/Java/Python (Open source compiler IDEs)

b) ***Assignment will have to be carried out by teaming up with one more team mate.***

c) Submission will have be to done thru a Google form on or before deadline.
Summary report will have to be handed over in ***hard copy to the co-faculty***

d) Approx 2-weeks of time will be available before submission. Actual dates will be broadcast. Hence look out !

e) Follow fair code of ethics and , **develop your own version** of the code. Plagiarism is to avoided.

f) Your team will be called upon to demo the assignment, to match with submission data you have provided in the Google forms /Hardcopy.

Problem Definition, Data Generation, Testing and Logging Stats

In the Text file (AESOP TALES.txt) provided as data

Develop implementations for the interface spec below:

Find_Length_of _Text(***txtfile***) // normalize multiple blank chars to
// single blank char and remove(store)
// website URLS that have infected
// text file using FSA based RegEx
// matcher

Find_Pattern (***pattern* , *InTextRange*, *algo***)
// Find the number of occurrences of
// ***pattern*** using any one of the
// following ***algo***rithms (2nd parameter)
// Rabin-Karp, Knuth_Morris_Pratt
// Suffix Tree (with Suffix arrays & LCP)
// ***InTextRange :*** can be indices or
// two patterns (e.g two story titles)

Build_Cross_Index(***txtfile, algo***)// Build an Index (Lex sorted)
// (**Word**, Number of occurrences,
// List of Story Titles & # of
// occurrences of ***Word)***

Find_Maximal,Palindromes***(PalindromeSize, InTextRange)***
// List maximal palindromes of size
// greater than or equal to
// ***PalindromeSize,*** with occurrences
(Story titles and indices)

Print_Stats ( ) // Text Size used, URL infection list found,
// Algo Used, Preprocessing time, Search time
// (Vary the parameters ***pattern ,***
// ***InTextRange)*** for ***timing plot***
// and Self Test & Verification outcome

Provide a simple command line interface to give a demo.

– Your observations/Learning outcomes about the assignment

FAQs:

1. What’s the Deadline and how to submit the assignment?
    a. Deadline: 18/Oct/2016. Be ready to demo your assignment to us and keep a copy of it in a memory stick, which we’ll copy.
2. I would like to team up with a friend in the other section (sections: B201 and B202) or else I would like to do the project alone.
    a. Kindly avoid that. Team up with someone from the same section. Learn to work with a teammate, find someone from the same section. If you are not able to find a teammate, share your name with the co-instructor (B201: Rashmi madam, B202: Divya madam). They will arrange a meeting with all those who don’t have a teammate and you’ll get a teammate in the meeting.
3. Could you let us know more about the format of the input file?
    a. The input file is an ASCII text file named “AESOP TALES (with random URLs).txt”. It has stories and each story has a distinct story title. A story title always starts after at least three new lines (some new lines may have some useless spaces). The file is infected with some URLs, which has nothing to do with the stories. You are expected to remove all the URLs and store it elsewhere. There are 20 URLs (I hope it helps you to make sure you are catching all the URLs). Then remove unnecessary blank spaces (you may not want to remove blank lines at this point because they help in identifying the story titles). At this stage, count the number of characters, identify the story titles.
4. New lines are affecting us to find patterns when the expected matching substring is split by a new line, but the pattern given doesn’t have a new line. How to deal with it?
    a. Handle the case carefully. The pattern typically won’t have new lines, but the matching substring may have a new line in between. You may want to treat the new line as a ‘space’ in your index.
5. There is just a list of functions, but how to execute them?
    a. The core of the assignment is about implementing a set of APIs. To demo the APIs you’ve implemented, write a program with a main function, which makes use of all the APIs and give us a demo of your APIs.
6. What is InTextRange in the API “Find_Pattern (***pattern* , *InTextRange*, *algo***)” ?
    a. The pattern to be searched in the range. The range could be a two indices indicating terminals of the range. The range could also be two distinct patterns. They could be story titles or any other substrings. Normally, they are distinct and first one appears before the second one. Handle abnormal cases too. If they appear in reversed order, swap them so that they are in proper order of first pattern appearing before the second one. If one or both patterns are not found in the text, the search is on a null range. If a pattern appears more than once, consider the first occurrence.
7. Should we just implement any one of the algorithms for “algo” in the API “Find_Pattern (***pattern* , *InTextRange*, *algo***)” ?
    a. No, you are expected to implement all the three algorithms. The user of the API (the demo program of your’s), calls the API with any one of the algorithms.