

UE14CS311 – Advanced Algorithms (5th Sem Elective)

Assignment 2 : String Search and Suffix Trees

1. Guidelines:

- Code can be developed in any of C/C++/Java/Python (Open source compiler IDEs)
- Assignment will have to be carried out by teaming up with one more team mate.***
- Submission will have to be done thru a Google form on or before deadline.
Summary report will have to be handed over in ***hard copy to the co-faculty***
- Approx 2-weeks of time will be available before submission. Actual dates will be broadcast. Hence look out !
- Follow fair code of ethics and , **develop your own version** of the code. Plagiarism is to avoided.
- 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 algorithms (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

