Project Document For

**String Matching: Plagiarism Detection**

**Submitted by: Debalina De(1726129)**

Knuth-Morris-Pratt Algorithm

KMP Algorithm was conceived in 1974 by Donald Knuth and Vaughan Pratt, and independently by James H. Morris. The three published it jointly in 1977.  The basic idea behind KMP’s algorithm is: whenever we detect a mismatch, we already know some of the characters in the text (since they matched the pattern characters prior to the mismatch). We take advantage of this information to avoid matching the characters that we know will anyway match. A matching time of O(n) is achieved by avoiding comparisons with elements of ‘S’ that have previously been involved in comparison with some element of the pattern ‘p’ to be matched. i.e., backtracking on the string ‘S’ never occurs. KMP algorithm does some preprocessing over the pattern and constructs a partial match table .For each sub pattern, it calculates the length of the maximum matching proper prefix which is also suffix of the sub – pattern and stores the value in the partial match table.

In this project, the algorithm is implemented in **JAVA.** The pattern file is split into sentences. Each such sentence is treated as a pattern, and is searched in the files present in the given directory. If the number of sentences found in the files inside the given directory is more than a certain threshold, the algorithm classifies the pattern file as plagiarized. Pseudo code of the code of the algorithm is given below –

1. Reads the pattern file and text files from the directory and store the text in multiple string variables
2. The String is then split based on the Full stop (.) and stored in a string array
3. Proper Prefix and Proper Suffix is calculated for the given pattern and stored in the array
4. Partial Match Table is constructed for each sub pattern and count of identical prefix and suffix values are calculated and stored in the table.
5. Each text file from the directory is compared with the pattern file, each sentence from the pattern file is compared with the text file ,when the sentences matches we increment the counter else if mismatch occurs we use the partial table values to skip ahead of the characters

Table [partial\_match\_length] > 1

we may skip ahead partial\_match\_length - table[partial\_match\_length - 1] characters.

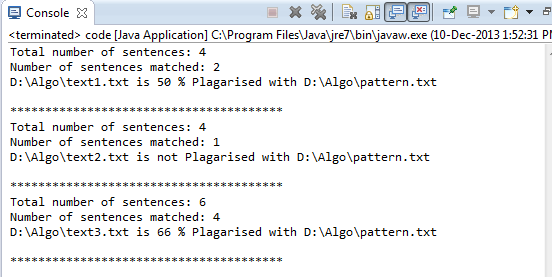
1. If the count(matching sentences) > Threshold Value

We declare that the text file is plagiarized with the pattern file

Else

The Text file is not plagiarized with the pattern.

**OUTPUT:**



To compare the performance of the algorithm in various cases, we have plotted the graph of number of iterations the algorithm needs to search a pattern to the varying lengths of the patterns. The graph is shown in following figure –

Figure – Performance of Knuth-Morris-Pratt Algorithm for varying pattern lengths and text lengths