Construction of Suffix Array for string Matching using C++

Hrishikesh V.
Department of Computer Science
PES University
Bangalore
PES1201700276

***Abstract*—Suffix arrays are used to store the suffixes of a string. Rather than storing the strings, they store the indices of the first character of the suffix in lexicographical order. If there are more than one string, the suffixes of each string are stored in a different suffix array.**

***Index Terms*—String Matching, Longest Common Prefix, LCP Array, Suffix Array**

I. Introduction

Suffix arrays are used to store the suffixes of a string. Rather than storing the strings, they store the indices of the first character of the suffix in lexicographical order.

If there are more than one string, the suffixes of each string are stored in a different suffix array. There is no need to end the strings with unique characters or to concatenate strings for a single suffix array. The time complexity for constructing a suffix array is o(n).

II. Construction

A loop is used to iterate through the string one character at a time. The indices are added to the suffix array and possibly the actual suffix starting at that index. Constructing the suffix can be done in constant time.

The second step in constructing the suffix array is to sort the indices of the suffixes in lexicographical order. Strcmp is called on the suffixes, which performs lexicographical comparison. The result is used to reorder the suffixes.

III. String Matching

A. Full Match

strncmp is used to compare the suffix and the pattern. An array stores the indices of all the matches. If this array is not empty, the first occurrence is returned.

B. Partial Match

Handling partial matches is more complicated than handling full matches. A combination of LCP Arrays and Suffix Arrays is used to find the longest common substring. This algorithm has a time complexity of O(2n+m) where the time taken to construct the suffix array for the text is O(n) and the time taken to construct the LCP array for the concatenated pattern and string is O(n+m).

To construct the LCP array, the two strings are first concatenated and a suffix array is constructed for the two. LCP array stores the longest common prefix of the strings present at adjacent indices.

IV. Output

In case of a full match, the first full match in all the documents is returned.

In case of a partial match, the index of the longest substring that first matches in a document is printed. If there are no partial matches or full matches, the function returns -1.

V. Conclusion

Suffix Arrays are faster to construct as opposed to generalized suffix trees using Ukkonen's Algorithm. However, Suffix Arrays are slower than trees but have better space complexity.