University of Glasgow | Department of Computing Science

Algorithmics

Assessed Exercise

Suffix Tree Applications

State & Strategy

Author:
Garry Sharp
0801585s

Supervisors:
Dr. D. Manlove

February 14, 2013

Contents

1 Task 1 (Search Suffix Tree)

The task here is implement a method *searchSuffixTree* in the class *SuffixTreeAppl* which searches over a global instance of a *SuffixTree* known as t with the search term being an array of bytes x

1.1 State

My solution works in an effective manner to search the tree in *O(n)* time. It should be noted that the solutions to Task 2 2, Searching for all occurrences relies heavily on this method. This means that there is a very small amount(almost negligable) amount of redundant code when searching (sets a global variable).

1.2 Solution

Firstly, lets examine the variables that are used. I keep track of an x index (initialised to 0) a node index (index of the characted being compared at the node), a boolean called match (which is set to true/false depending on whether there is a match), an integer start location (initialised to -1) as well as a SuffixTreeNode called currentNode (initialised to the child of the root of the tree).

My solution comparing the relevant values of x with the relevant values held at the node. If all characters are matched successfully (and there are more in subsequent nodes) then I perform the same check at the child of that node. If all characters have been matched and no more are left then the loop terminates.

If there is a mismatch at any characters then I examine the sibling that node until, either there is another match (is a mismatch occurs at that node it looks at the sibling, otherwise if fully matched goes to the child) or a mismatch at which point the loop exits.

A variable t2Node is set throughout and this is so that Task 2 can be completed by recycling code, this node is eventually set to the final node that contains leafs of the instances of the search term.

If a match does occur and startLocation is not already set, a variable startLocation is set to it's left label of the currentNode. Otherwise is a mismatch occurs startLabel is reset (set to -1).

The program terminates by setting the Task1Info object with position startLocation (which will be a -1 if not found) to the main method which then displays it accordingly. The search executes in O(n) time.

2 Task 2 (All Occurrences

2.1 State

2.2 Solution

3 Task 3 (Longest Repeating Substring)

3.1 State

3.2 Solution

4 Task4 (Longest Common Substring)

4.1 State

4.2 Solution