2374535C Algs 2 report

Disclaimer: command line java had some issues on my working computer tested this all with eclipse configurable variables and a friend ran code from command line to make sure it worked(it did) if theres any issues when running command line for this im sorry

Tasks 1-3 working fine

Task 4 method not working. data structure and main implemented fine causes no crashes –

How I’m supposed to do this didn’t click at time of hand in, issues are implementing the computation of the b_i values during traversal, but I believe the implementation of the path label to be almost correct

Task 1

For task 1 I used the inbuilt insert function changing it so that instead of adding a new suffix when found, it would return the position once the length of the path exceeded the length of the search string

Checks if a node has a child with a left edge label that matches the first character in the search string, if so checking each character matches until the right edge label is reached, keeps track of total path length

Sets a task1info.setmatchnode to next for use in task 2

Task 2

Changed the main method of traversing the suffix tree to recursion, the implementation uses task 1 to get the node where the search string matches and using its child as a starting point to count leaves. From there as long as the string exists, the program will travel down each child until reaching a leaf, adding the position to a linked list in the task2 object. Once children are exhausted siblings are checked the same way.

Task 3

Uses same recursive method as task 2 to traverse the tree but with an extra variable too keep track of depth, each time the method is called to go down to a child string depth is updated with the difference between edge labels +1 whenever a leaf node is reached a check is made if it has a sibling and therefore if the substring is repeated, doesn’t check for repeats after the first but could check for more siblings. If the current string depth is more than the one stored in the task3 object then the positions of each suffix is stored with the new depth.

Task 4

Same recursive method to traverse as in 3

Implementation is just task3 slightly changed since not working cant really argue for it or how efficient it is