As usual, show your work.
Due March 8th, 2022, 11:59PM.

Problem 1

The bus

[2 points]

We have following collection of strings called Patterns and a string called Text.
Text := tactnahhctndhhctna
Patterns :=

1. tah
2. acta
3. hhctnd
4. ca
5. hhca
6. tatt
7. tac
8. actna
9. hhctna
10. ctn

Construct *Trie(Patterns)* using the algorithm we learned in class. How many leaves does the trie have?

Problem 2

Bus stops

[3 points]

Perform TrieMatching between *Text* and *Trie(Patterns)* you generated in question 1 according to the algorithm we learned in class. That is, find all substrings of *Text* that match any strings in Patterns. Show all iterations of the TrieMatching algorithm; at each iteration, show the input and output by TrieMatching. How many matches were found? That is, how many substrings of *Text* matched one of strings in Patterns?

Problem 3

Suffix trees

[5 points]

Create a suffix tree of *Text*. Show all suffixes of *Text* first and draw the tree. How many leaves are in the suffix tree? Is it the same as the number of leaves in Trie(Pattern)? Why or why not?

Problem 4

Suffix arrays

[2 points]

Create a suffix array of *Text*. Show sorted suffixes and their starting positions.

Problem 5

BWT

[5 points]

Create the Burrows-Wheeler transform of *Text*. First, create cyclic rotations matrix. Second, create M(*Text*) matrix. Lastly, output BWT(*Text*) from M(*Text*).

Problem 6

BWT traversal

[5 points]

Reconstruct the string whose Burrows-Wheeler transform is *tttttacg*$*gacaacc*. Show how you reconstruct each letter using the First-Last property in each iteration as we learned in class (refer to Figure 9.12 in textbook). In other words, show the partial M(*Text*) matrix and the two arrows as in Figure 9.12 to indicate each letter you are reconstructing. Be sure to label the edges according to the order of how they should be traversed.

Problem 7

BWMatching

[8 points]

Using the M(*Text*) matrix created in question 5, perform BWMatching between *Text* and the following 2 strings in Patterns. First, create a table that has i, FirstColumn, LastColumn, and LastToFirst(i) as in Figure 9.15 in textbook (or as in lecture slides). Then, for each string

in Pattern, show the values of top and bottom variables in each iteration of BWMatching, as in Figure 9.14 in textbook. How many times does each string in Pattern appear in *Text*?

Patterns:

1. ctn
2. cna