COT 6417
Algorithms on Strings and Sequences
Fall 2020
Homework Assignment 1

Please complete this assignment individually (and not in a group). Please email me a single pdf document with the solutions (*only pdf is accepted*) by *September 22, 2020*. Also, you need to only provide one solution to a problem.

1. Given two strings A and B, of lengths n and m respectively, describe an $O(n + m)$ time algorithm that finds the longest suffix of A that exactly matches a prefix of B.

2. Let T be a text string of length m and let S be a multiset of n characters. The problem is to find all substrings of T of length n that are formed by the characters of S. Note that, for this problem, the order of the characters from S that appear in T does not matter. So, for instance, if T = *aabxyaba* and $S = \{a,a,b\}$, then both substrings *aab* and *aba* fit the solution. Provide an algorithm for this problem that runs in $O(m)$ time. Assume that the alphabet is of constant size.

3. Let T be a string whose characters are drawn from the alphabet Σ. We are given three strings α_1, α_2, and α_3, all of whose characters are also drawn from the same alphabet. Let P be the pattern obtained by concatenating the three strings in order, but with two '*' characters inserted between each α_i and α_{i+1} ($1 \leq i \leq 2$). The '*' character is called a wild card character and can match any character in the alphabet Σ. Thus, pattern P is of the form α_1**α_2** α_3. The problem is to determine if P occurs in the text T. Provide a linear time algorithm for this problem.

4. For a string S of length n, recall that $sp_i(S)$ is defined to be the length of the longest proper suffix of $S[1..i]$ that matches a prefix of S. Recall also that $sp_i'(S)$ is defined to be the length of the longest proper suffix of $S[1..i]$ that matches a prefix of S and with the added condition that $S(i+1) \neq S(sp_i' + 1)$. For a string S, (a) given its sp_i values for all $1 \leq i \leq n$, provide an $O(n)$ algorithm to compute its sp_i' values for all $1 \leq i \leq n$; (b) given its sp_i' values for all $1 \leq i \leq n$, provide an $O(n)$ algorithm to compute its sp_i values for all $1 \leq i \leq n$.

5. (See $sp_i(S)$ definition from above problem). You had written down a 11-bit password S on a piece of paper but have now lost the paper. However, you recall a few facts about the password S: that the first bit was a 1, and that $sp_{11}(S) = 6$, $sp_6(S) = 3$, and $sp_2(S) = 0$. Can you reconstruct the password S? Explain your reasoning.

6. (McCreight's suffix tree construction method). Let S be the string *MISSISSIPPI$*. Recall McCreight's suffix tree construction method; let ST_i denote the suffix tree containing the first i suffixes of S (i.e. all strings $S[j,m]$, for $1 \leq j \leq i \leq m$, and where m is the length of S). Show the suffix tree ST_i at the end of each iteration i, for $1 \leq i \leq 12$. Please be sure to include the suffix links as well.