Homework 1

CS 4390/5390
Fall 2019

Due: 18 September 2019

This homework is worth 5 points out of the total 25 points of homework in the class.

1. **(2 points)** Give an algorithm that takes in two string α and β, of length n and m, and finds the longest suffix of α that exactly matches a prefix of β. The algorithm should run in $O(n+m)$ time.

2. **(2 points)** Given two strings of length n characters each and an additional parameter k. In each string there are $n-k+1$ substrings of length k (we will later call them k-mers), and so there are $\Theta(n^2)$ pairs of substring, where one substring is from one string and one is from the other. For a pair of substrings, we define the *match-count* as the number of opposing characters that match when the two substrings of length k are aligned. The problems is to compute the match-count for each of the $\Theta(n^2)$ pairs of substrings from the two strings. Clearly the problem can be solved with $O(kn^2)$ operations (character comparison plus arithmetic operation). But by better organizing the computations, the time can be reduced to $O(n^2)$ operations. (Note, this problem can be solved without the Z algorithm or other fancy tools, only thought.)

3. **(1 point)** Given a set $\mathcal{S}$ of k strings, we want to find evey string in $\mathcal{S}$ that is a substring of some other strings in $\mathcal{S}$. Assuming the total length of all of the strings is n, give an $O(n)$-time algorithm to solve this problem.