02-714: Homework #3

Due: Oct. 29 at the start of class

Please write your answers neatly or typeset them. You may discuss the problems with your current classmates, but you must write your own solutions entirely independently. If you need to make any assumptions in order to solve a problem, state them explicitly. If you consult any sources, cite them. Please do not provide complex pseudocode — describe your algorithm in a way that is readable for a human.

1. Let S be a string of length n. Give an $O(n)$-time algorithm to find the longest repeated substring of S such that at least two copies of the substring do not overlap in S.

2. A *k-mismatch palindrome* is a string xy where, $|x| = |y|$ and reverse(y) and x are the same in all but at most k positions. Give an $O(kn)$-time algorithm to find all the k-mismatch palindromes in a string S of length n.

3. You are given an ordered collection of documents $D_1, \ldots, D_t$ that represent some main document as it changed over time ($D_{t'}$ is the document at time t'). You can treat each of the documents as a list of words.

   Define $\texttt{span}(w)$ to be (i, j) where i is the first document in which w appears and j is the last. Show how to preprocess $D_1, \ldots, D_t$ into a wavelet tree so that you can find span(w) for any w in $O(\log n)$ time.

4. (Gusfield) Let x be a string of length n. There are $O(n^2)$ substrings of x. Show how to count the number of *distinct* substrings of x in $O(n)$ time.

5. (Gusfield) Let x and y be two strings, each of length n. Let D be the set of substrings that are in x but not in y. Give an $O(n)$-time algorithm to construct an Aho-Corasick-like keyword tree for the strings in D.

6. Recall that the zero-order empirical entropy of a string S is:

$$H_0(S) = \sum_{c \in \Sigma(S)} p_c \log(1/p_c), \tag{1}$$

   where p_c is the fraction of times character c appears in the string S and $\Sigma(S)$ is the set of characters that appear in S.

   Let S be a string of length n. Show that if the bitmap B_u at each node u in a complete, balanced wavelet tree for S is compressed to take $|B_u|H_0(B_u)$ bits, then the total number of bits used by the bitmaps would be $nH_0(S)$ which would be the bits used in an optimal zero-order encoding of S. (Note that the B_u's are over a binary alphabet and S may be over a larger alphabet. You can assume that $|\Sigma| = 2^i$ for some i.)