CSC 226 SUMMER 2018
ALGORITHMS AND DATA STRUCTURES II
ASSIGNMENT 4 - WRITTEN
UNIVERSITY OF VICTORIA

1. Show the longest common subsequence table, `llcs[][]`, for the following two strings:

   x = "skullandbones"
   y = "lullabybabies"

   What is the longest common subsequence between these strings?

2. Write a pseudocode description of the `printLCS()` algorithm, which prints the longest common subsequence of two strings x and y. Your algorithm should take as input the completed `llcs[][]` integer array of longest common subsequence lengths, and the two strings x and y. (So, you do not have the `path[][]` array - see Lecture 19, slides 100 and 101.) Your algorithm must return the specific string corresponding the length found in `llcs[n][m]` and it should run in $O(n+m)$ time, where n is the length of x and m is the length of y.

3. (a).Give the right[] array computed by the Boyer-Moore algorithm for the pattern **ABRACADABRA**. Assume the alphabet is $\Sigma = \{A, B, C, D, R\}$.

   (b). Give the `dfa[][]` array for the KMP algorithm for the pattern **ABRACADABRA**, and draw the transition diagram corresponding to it. Assume the alphabet is $\Sigma = \{A, B, C, D, R\}$.

4. Write an efficient method that takes a string txt and an integer M as arguments and returns the position of the first occurrence of M consecutive blanks in the string. If there is no such occurrence it should return the length of txt (txt.length). Give a runtime analysis of the method.

5. Modify the Rabin-Karp algorithm to search for a given pattern with the additional provision that some character in the pattern is a wildcard, that is, it can be any character. So, for some $k = 0, 1, \ldots, M - 1$, where the length of the pattern is M, the kth character of the pattern can be any character. You may assume that k is given as input along with the text and pattern.