605.621 Foundations of Algorithms Fall 2022
Programming Assignment #4
Assigned with Module 12, Due at the end of Module 14

Due to requirements for submitting grades to the registrar, PA4 cannot be submitted late. PA4 submissions made after December 12 @ 11:59 P.M. Eastern time will not be graded.

The goals of Programming Assignment 4 are: (1) to practice your algorithm design and specification, (2) to practice your trace runs and tests to measure asymptotic behavior, and (3) to practice analysis of theoretical asymptotic behavior compared to actual program asymptotic behavior.

In this course, critical thinking and problem analysis results in discovering appropriate and better, the best, algorithm for a problem; defining the algorithm in pseudocode; demonstrating (we don't usually prove) that the algorithm is correct; and determining the asymptotic runtime for the algorithm using one or more of the tools for asymptotic analysis you have studied this semester.

This programming problem is not a collaborative assignment. You are required to follow the **Programming Assignment Guidelines** and **Pseudocode Restrictions** (Canvas page *Pseudocode Restrictions & Programming Guidelines*) when preparing your solutions to these problems.

Exercising the algorithm you design using one or more example data set(s) removes any doubt from the grader that the algorithm is structurally correct and all computations are correct. When a problem supplies data you are expected to use it to demonstrate that your algorithm is correct.

You are permitted to use Internet resources while solving problems, however complete references must be provided. Please follow the Sheridan Libraries' citation guidance at "*Citing Other Things* - HOW DO YOU CITE AN INTERVIEW, A TWEET, OR A PUBLIC WEB PAGE?" and continue to the APA Academic Writer *Sample References*. Additional example citations are provided at the Purdue University, Purdue Online Writing Lab (OWL), College of Liberal Arts *Reference List: Electronic Sources*. You can also use training provided by Victoria University Library, Melbourne Australia on *Harvard Referencing: Internet/websites*

Paragraph 1 You are consulting for a group of people whose job consists of monitoring and analyzing electronic signals coming from ships in the Atlantic ocean. They want a fast algorithm for a basic primitive that arises frequently: "untangling" a superposition of two **known signals**.

Paragraph 2 Specifically, they aresolving a situation in which each of two ships is emitting a short sequence of 0s and 1s over and over, and they want to make sure that the signal, s, they are receiving is simply an *interweaving* of these two emissions, with nothing extra added in. **The signal, s, is received as a stream of symbols by the you, the receiver. You must process the signal in the order you receive the symbols; you cannot receive all of s and then operate your algorithm on the fully received s.** The short sequence emitted by each ship is known to the ship and the receiver. The only symbols that can be received are in the alphabet $\Sigma = \{0, 1\}*$, including any symbols received that are not in x and y. Symbols received could contain symbols due to noise in the transmitter/receiver frequency or symbols injected into s by a bad actor.

Paragraph 3 Definition: Given a signal x consisting of 0s and 1s, we write x^k to denote k copies of x concatenated together. We say that signal x' is a *repetition* of x if it is a prefix of x^k for some number k. So, $x' = 101101101$ is a repetition of $x = 101$.

Paragraph 4 We say that a signal s is an *interweaving* of x and y if its symbols can be partitioned into two (not necessarily contiguous) subsequence s' and s'' so that s' is a repetition of x and s'' is a repetition of y. Each symbol in s must belong to exactly one of s' or s''. For example, if $x = 101$ and $y = 0$, then $s = 100010101$ is an *interweaving* of x and y since characters 1,2,5,7,8, and 9 form 101101 – a repetition of x – and the remaining characters 3,4,6 form 000 – a repetition of y. In terms of our application, x and y are the repeating sequences from the two ships, and s is the signal we are receiving.

Paragraph 5 You want a "fast algorithm for ...'untangling' a superposition of two known signals", x and y. You receive some set of symbols s. **You do not know if the signal you receive has its first symbol in x, y, or symbol from noise or a bad actor.** You need to use your knowledge of x and y and the received signal s to determine if s consists only of valid *interwoven* symbols from x and y, ignoring inital symbols that may be fragments of x, y, or noise/bad actor. That may require your solution to discard some symbols at the beginning or end of s to get to at least a full match of x and a full match of y. The received signal could be too short to do this.

(Assignment continues on the next page)

Assignment Points:

1. [30 points] Give an efficient algorithm that takes signals s, x, and y and decides if s is an *interweaving* of x and y. Output the assignment of symbols in s to to symbols in x, y, or noise. Derive the computational complexity of your algorithm.
   (a) [-10 points] If your algorithm assumes that s begins with either of the first symbols of x or y you have not fully solved the problem.
   (b) [-10 points] If your algorithm does not address symbols due to noise you have not fully solved the problem.
   (c) [15 points] Algorithm in pseudocode.
   (d) [15 points] Asymptotic analysis of the algorithm.

2. [30 points] Implement your algorithm and test its run time to verify your complexity analysis.
   (a) [15 points] Code implementing your algorithm
   (b) [5 points] Instrumentation and tests to measure the asymptotic behavior of your program.
   (c) [10 points] Comparison and analysis of algorithm and implementation asymptotic behavior

3. [40 points] **Required test cases**:
   Submit one text file containing solutions for all of the required test cases and results. For each of the required test cases you must submit: **[(i) the test case identifier {*s1*, *s2*, *s3*, *s4*} (ii) the input for that test case, (iii) the final assignment of symbols in s to symbols in x, y, or noise, and (iv) whether s is an interweaving of ship x and y signals]**.
   (a) [10 points] Example in Paragraph 4 - Let *s1* = "100010101" and x = "101", y = "0". *s1* is an interweaving of x and y where {[1,2,5],[7,8,9]} are repetitions of x and {[3],[4],[6]} are repetitions of y.
   (b) In each of the following let x = "101" and y = "010".
      i. [10 points] Let *s2* = "101010101010101". *s2* is an interweaving of x and y where {[1, 2, 3], [7, 8, 9], [13, 14, 15]]} are repetitions of x and {[4, 5, 6], [10, 11, 12]} are repetitions of y
      ii. [10 points] Let *s3* = "001100110101011001100110010101111". *s3* is in interweaving of x and y, ignoring leading noise and trailing noise, where {[12, 13, 14], [15, 16, 18], [23, 25, 26]} are repetitions of x and
      {[9, 10, 11], [17, 19, 20], [21, 22, 24],[27, 28, 29]} are repetitions of y, and
      {[1, 2, 3, 4, 5, 6, 7, 8],[30, 31, 32, 33]} are noise.
      iii. [10 points] Let *s4* = "100110011001". *s4* is an interweaving of x and y where {[1,2,4], [9,11,12]} are repetitions of x and {[3,5,6],[7,8,10]} are repetitions of y.
   (c) Alternative solutions for these required test cases will not be entertained. To be fully correct, your algorithm must produce the solutions provided for each of the four required test cases.

NOTE: The assigned problem is not an optimization problem. The problem does not request the maximum or minimum of any attribute. Any solutions using optimization will be assigned a grade of 0. Solutions based on the following Internet resources, for example, will be assigned a grade of 0:

1. *cs.stackexchange.com question 121459 solved with dynamic programming* or
2. *GeeksforGeeks Find if a string is interleaved of two other strings — DP-33* or
3. *How to Solve Sliding Window Problems: An Intro To Dynamic Programming*

STREAM OF SYMBOLS: Think fluid leaving a faucet. At any point before, at, and after the terminus of the faucet you can examine the data at that point. Before that point the data have not yet arrived, after that point the data are no longer available, and at that point the data can be examined. You might refer to paragraph two of Chapter 27 Online Algorithms, on page 790 of CLRS. Algorithms that process s offline rather than online will be assigned a grade of 0.