Programming Assignment 4

# Overview

The objective of this programming assignment is to decode an incoming signal, , that consists of intertwined repeating signals, and , from two separate different ships, along with filtering out potential noise either before or after the signal. The proposed algorithm will take , , and as an input, and output the indices in that make up instances of and , as well as the indices that contain unexpected noise.

# Algorithm

## Approach to Implementation

To start out, this algorithm can be seen as an overarching loop iterating over the incoming signal, , where on each iteration the current symbol is properly handled depending on the status of the overall signal. This can break down into several cases, handling if the current symbol in matches the current symbol of and , only , only , or neither nor . In such cases we can determine where the current symbol is best fit to determine if is an interweaving of and . Once a high-level algorithm was completed, edge cases could be handled, such as the addition of handling noise or covering specific cases in determining which signal the current symbol should go towards in the case that either signal could work. Luckily, given specific test cases allowed the algorithm to be fine-tuned to match exactly the desired output of the test cases, specifically in these edge cases, this can be seen in lines 9 - 13 of the pseudocode.

## Asymptotic Analysis

As described above, and seen below, the algorithm iterates over to mimic an incoming signal being read as if it is being received. This causes an overarching for-loop to execute times, where is the number of symbols in . Although there exist cases when noise must be handled using two additional for-loops, these will execute at most the size of either signal and , which will note increase the overall asymptotic behavior. For these reasons, the asymptotic behavior of this algorithm is , where is the number of symbols in .

The actual and expected key points are shown when executing the program, they can be seen in TestCasesOutput.txt.

## Pseudocode

