FIT3155 - Assignment 1

Name - Sadeeptha Bandara
ID - 30769140

Comments to the marker: Q1

This implementation behaves similarly to if the pattern was appended to both the front and end of the reference text and if the z arrays were computed accordingly

pat$ref$pat

The z prefix and z suffix values can be computed for **each index in the reference text** as per the above setup using a standard z algorithm.

Then each m length region of the reference text is considered (where m is the length of the pattern) and if m - (z_i + z_suffix_i) = 2 and if the characters swapped were a match, it would detect a transposition in the text.
The complete matches are found by just checking cases where z_i = m

However, the implementation makes use of a number of optimizations to implement the above logic efficiently

1. Uses a basic z algorithm on the pattern to compute the z prefix and z suffix for the **pattern** only. This takes away the need to slice the array later, to separate the z values for the reference only.
2. Then uses this information to compute the z array for the prefix and the suffix of the **reference** with respect to the pattern in a single loop. Saving time and space
3. The reference text is looped over and for every m letters, it is checked if
   Z_i =m (corresponding to a full match)
   m - (z_i + z_suffix_i) = 2 (Only two characters mismatch) and if the mismatched characters match if swapped, this would correspond to a transposition.

The implementation uses O(n+m) space and time, where n is the length of the reference text and m is the length of the pattern.