FIT3155: Lab questions for week 3

Objectives: This prac allows you to explore the concepts learnt in weeks 1 and 2

1. From last week's lab, you should have a working implementation of Z-algorithm that computes the Z_i values for any given string. You will need this when implementing Boyer-Moore and Knuth-Morris-Pratt's algorithms. If not complete, now is the time to complete your implementation.

2. Implement Boyer-Moore's algorithm.

   - Your implementation MUST use the Z-algorithm to preprocess `pat` for `goodsuffix`(.) and `matchedprefix`(.) values.
   - Also, ensure that after each shift when implementing the good suffix rule, there are potentially some character(s) in the `pat` that already match the `txt`. Your implementation should not compare these again (unnecessarily).

3. Implement Knuth-Morris-Pratt's algorithm.

   - Your implementation MUST use the Z-algorithm to preprocess `pat` for $\mathbf{SP}_i$ values.
   - Also, after each shift, there are potentially some character(s) in the `pat` that already match the `txt`. Your implementation should not compare these again (unnecessarily).

4. Refer to prac resources for week 3 made available to you on Moodle. The file `reference.txt` contains a long string. Another file `pattern-collection.txt` contains a set of 100 patterns (one per each line of the file).

   - Find all occurrences of each of the given 100 patterns in the reference string using your Boyer-Moore implementation.
   - Find all occurrences of each of the given 100 patterns in the reference string using your Knuth-Morris-Pratt implementation.
   - Compare the total runtime for searching for this collection of patterns against your Z-algorithm implementation from last week.

5. In preparation for next week's lab, implement the naive $O(n^2)$-time method of constructing a *suffix tree* of any given string `str`$[1..n]$. This would be a good practice to understand suffix tree construction before implementing Ukkonen's $O(n)$-time algorithm (next week).

```
-=0=-
 END
-=0=-
```