FIT3155: Week 4 tutorial
Covering concepts from Weeks 2 and 3

Objectives: The tutorials, in general, give practice in problem solving, in analysis of algorithms and data-structures, and in logic useful in the above.

Instructions to the class: Prepare your answers to the questions **before** the tutorial. It will probably not be possible to cover all questions unless the class has prepared them all in advance.

Instructions to Tutors:

i. The purpose of the tutorials is not to solve the practical exercises.

ii. The purpose is to check answers, and to discuss particular sticking points, not to simply make answers available.

1. Obtain the suffix trees and their corresponding implicit suffix trees for the following strings:
    - `abcdefg`
    - `aaaaaaa`
    - `abcabc`
2. Identify the suffix links in the implicit suffix tree for the string `aabcaba`.
3. Reason how a suffix tree of a string S could be used to answer the following problems:
    - Given a string t, find if t is a substring of S.
    - Given a string t, find if t is a suffix of S.
    - Find the number of times a string t occurs in S as a substring.
    - Find the smallest substring of S occurring exactly k times.
    - Find the longest repeated substring within S.
    - Find the lexicographically smallest suffix of S
    - Find the suffix array of string S.
4. Revise Ukkonen's algorithm for computing suffix trees.

5. In Boyer-Moore algorithm, as discussed in the lecture, the scanning is performed right to left in the pattern, although the pattern moves left to right relative to the reference text.

   If we instead started by aligning the rightmost ends of the pattern and text, and conducted a left-to-right scan in the pattern, and shifting the pattern left after mismatch, can you work out the details of this Boyer-Moore variant for pattern matching.

-=o0o=-
END
-=o0o=-