FIT2004: Lab questions for week 7

Objectives: This prac provides a platform for you to learn the formal concepts introduced during lectures in weeks 5& 6. Primarily, these concepts include the construction and basic operations on integer and string search and retrieval trees. NOTE: This prac is **NOT** assessed. You are free to write programs in Java or Python, whichever you find convenient.

If you have not finished the Trie question from last week's lab, please do it now.

1. Write a program to generate a suffix trie of a given string `str` (e.g., `mississippi`) by inserting all of its suffixes in the tree. Then, implement functions to support following queries.

   (a) Given an input substring `sub`, find whether `sub` is a substring of `str` or not. E.g., `iss` is a substring of `mississippi` but `isp` is not.

   (b) Count the number of occurrences of a substring `sub` in `str`. E.g., `si` occurs two times in `mississippi`.

   (c) Find the longest repeated substring in `str`. E.g., `issi` is the longest repeated substring in `mississippi`.

2. Write a program to construct an AVL tree from a set of distinct integers. Your program should support `search`, `insert` and `delete` operations while always ensuring that the AVL tree structural property is maintained. Test your program on (starting with) the exercise given in this weeks Tute sheet.

```
-=o0o=-
  END
-=o0o=-
```