AVL & Dynamic Programming

Advanced Algorithms – Assign#3 **Due date: Oct. 13th**

1) Show the result of inserting the nodes 13, 8, 5, 9, 4, 6, 12, 2, 1 and 3 in the order given to an initially empty AVL tree. Show your steps after each insertion. It is not enough to just show the final output.

2) Write pseudo code that verifies that an AVL tree is a valid AVL tree. Your algorithm should verify (i) Tree is a binary search tree, (ii) height stored in each node is correct, and (iii) each node maintains height-balance property. Optionally include code as well. Also include the time complexity of your pseudo code.

3)

15-2 Longest palindrome subsequence

A ***palindrome*** is a nonempty string over some alphabet that reads the same forward and backward. Examples of palindromes are all strings of length 1, `civic`, `racecar`, and `aibohphobia` (fear of palindromes).

Give an efficient algorithm to find the longest palindrome that is a subsequence of a given input string. For example, given the input `character`, your algorithm should return `carac`. What is the running time of your algorithm?

Also show the dry run of your algorithm labelling the steps.

Submission:
Please upload your solution files to d2l by the due date.