Department of Computing and Information Systems
COMP90038 Algorithms and Complexity Tutorial Week 10

Plan

This week we will try to keep up with the lecture plan (Levitin 7.1–7.3) but also find some lecture time to discuss the assignment.

The exercises

1. Give an algorithm for deciding whether an array $A[1..n]$ is a heap.

2. Construct a binary search tree (BST) by starting from an empty tree and inserting these keys, in the given order: 17, 19, 13, 26, 14, 18, 21, 24.

3. Construct an AVL tree from the empty tree by inserting the following keys in the given order: A, L, G, O, R, I, T, H, M.

4. Construct a 2–3 tree from the empty tree by inserting the following keys in the given order: A, L, G, O, R, I, T, H, M.

5. Consider the set of five keys (let us say they are positive integers) $\{k_1, k_2, k_3, k_4, k_5\}$, satisfying $k_1 < k_2 < k_3 < k_4 < k_5$. There are 120 different permutations of these five keys. For exactly two of the 120 permutations, the following happens, when the keys are inserted one by one, in the order given by the permutation, into an initially empty AVL tree: First an LR-rotation takes place, then an RL-rotation takes place. Which two permutations generate that behaviour?