Exercises Problem Set 2

Meta-Algorithm: Before you begin designing an algorithm or a proof it is important to ensure you have a good understanding the problem. Ensure you've read the description carefully and attempt to settle every doubt. A great strategy is to try to solve a small instance of the problem before attempting a full solution, i.e. try very small n (2 or 3 can be enough) or devise a toy-version of the problem that you can solve quickly to develop an intuition.

Question 1. A binary tree is useful structure for many applications, for intance it can be used to model a sequence of yes/no-questions where a left-node represents yes and right-node representing no.

a) Design a binary tree with questions as internal node (nodes with children) and the following set of characters as leaf nodes (nodes without children)

$$\{\text{Jan van Rijn}, \text{Einstein}, \text{Nemo (the fish)}, \text{Nijntje}\}$$

b) You might recognize this as the game 20-questions. How many things (i.e. leaf nodes) are possible in a tree where every branch contains 20 yes/no questions? Note that this is different from a total 20 yes/no questions, as the questions asked may depend on earlier answers.

c)* A binary tree is called *perfect* if all internal nodes have two children and all leaves occur at the same heigth. How many nodes are there in a perfect binary tree of heigth 20?[1]

Question 2. Given a binary tree, design a recursive pseudocode algorithm that returns the highest value of the tree. **Hint:** Use two functions.

Recall that a *Binary Search Tree* (BST) is a Binary tree in which satisfy the **BST-property**. We say that a binary tree T, satisfies the **BST-property** if, for **any** node $x \in T$ the following two conditions are true:

i. Every node in the left subtree of x, with value K, have value less than or equal to K.

ii. Every node in the right subtree of x, with value K, have value greater than or equal to K.

Question 3. Draw the binary search tree which is created by inserting the following numbers in the given order: 42, 53, 12, 16, 8, 2, 60, 57, 65, 22, 19

Question 4. With BSTs certain problems can be solved more efficiently, but there is a price to be paid in some additional complexity to ensure the BST-property is not violated. You may assume that all values in the tree are unique.

a) Design a pseudocode algorithm that returns the highest value in a BST.

b) Design a pseudocode algorithm for deleting the root node of a BST. The resulting tree should again be a BST.

Question 5. Solve this week's programming work sheet.

[1] Questions marked with a * are a more challenging