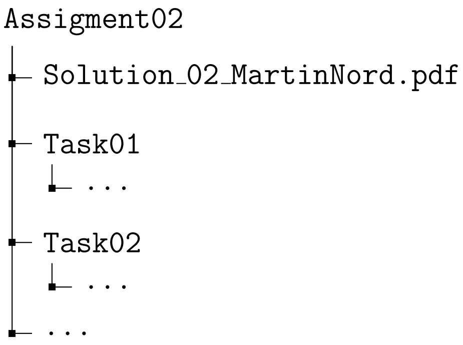

Figure 1: Exemplary directory structure

Assignment 2

Submission procedure

Your submissions must meet the following requirements:

1. Please organize your solution into a single tar or zip file.
2. Please submit your solution as pure Python files (.py).
3. Provide sufficient documentation and/or tests so that we can run your code.
4. Structure and work on your submission. For example:
   - Create a directory for the Assignment, e.g. name the directory "Assignment2" for Assignment 2.
   - Create a subdirectory for each task with the task number as name, e.g. "Task01" for task 1.
   - Place the solution within the respective subdirectory.
   - Solution to pen & paper tasks must be aggregated to a single pdf file. The file name is expected to follow the pattern be "Solution_AB_Name_XY.pdf", where XY must be substituted with your name and AB with the exercise sheet number.
   - See Figure 1 for an exemplary directory hierarchy.
5. Upload the solution before deadline.

Task 1 *Tree*

1 p.

Implement LCRSNode in python and add degree, is_leaf, size and height functions. [Slide 16]

Task 2 *Binary Trees*

2 p.

Implement a binary tree in Python and provide a function that traverses all nodes. [Slides 20ff] Hint: The two algorithms from the lecture both use recursion to visit all the nodes in a binary tree.

Task 3 *Binary Search Tree*

2 p.

Implement a binary search tree in Python and provide an example of an input sequence that leads to a degenerate tree [Slides 23ff].

Task 4 *Red-black trees*

7 p.

Implement a red-black tree in Python; you only need to implement the rb_insert function [Slides 36ff]. Show that the example given in the last task does not lead to a degenerate tree.

Task	1	2	3	4	total
Points	1	2	2	7	12
reached					