Data Structures 2018

Exercise 7 (Week 43)

- **Notice that based on university's new regulations on degrees, you can have a degree fail from the course which is put to the register.**

  > If a student does not participate in the course and does not cancel his/her enrollment, or if he/she discontinues the course, he/she will be assigned a fail grade for the course in question.

- **Students who participate in exercise group must be in place before the exercise group begins (12.15/14.15/16.15). Students who come late do not get the exercise points.**
- **Check the numbers of exercises made before you come to the exercise group. By this means we can save a lot of time when filling the exercise point list.**
- **Notice that pseudocode does not mean the same this as Java code. Pseudocode is not a programming language dependent presentation for an algorithm.**

1.-3. Typically a binary search tree is implemented according to the following definition: A binary tree is an empty tree or it consists of a root node with two children, left and right, which are binary trees. In the implementations the empty tree can be represented as the null-reference and any node can have 0, 1 or 2 children. According to this definition leaf nodes are the nodes with empty trees as **both** children.

The advantage of this definition is that there are no extra nodes. In fact a binary search tree implemented using the strict definition of the binary tree contains roughly half of the nodes (namely all leaf nodes) that contain no information relevant to the solution of the task.

Implement a dictionary using a binary search tree according to this non-strict definition, which contains elements of type Object stored with keys of type int, into the file `BinarySearchTree.java` and test it with the program `BSTTest.java`. You also need the file `BinaryTreeNode.java`.

Test 1 requires you to implement insert and find, test 2 requires you to implement deletion. The test program also contains a small test, which prints the results of the operations to make the development easier.

(Each implemented operation = 1 point)

4. What does the resulting heap look like, when we insert the keys 2, 6, 7, 8, 0, 3, 11, 9, 1, 10, 4, 5 into an empty heap in this order ?
5. Describe what happends in the heap of the previous assignment, when the smallest key is extracted three times.
6. Describe how a binary search tree is formed, when the keys 6, 4, 5, 11, 3, 10, 12, 1, 2, 7, 9, 8 are inserted to an empty binary tree in this order.
7. Describe how the tree in the previous excercise behaves when the keys 2, 5, 4, 6 are deleted in this order.