Data structures, exercise 5, 16.-20.2.

1. Insert to an empty binary search tree keys 2, 0, 3, 7, 9, 1, 5, 6 ja 8. Show how the tree grows when the insertions are made.

   After the insertions, show how the keys 3, 9, 2 and 5 are deleted from the resulting tree.

2. Define an algorithm that counts the number of nodes in a binary tree.

   What is the time and space complexity of the algorithm?

3. Define an algorithm that counts how many times a given key is in a binary tree. Eg. in the below tree, if the key to count is 3, the answer would be 2.

```
  3
 / \
2   5
   / \
  3   4
```

   What is the time and space complexity of the algorithm?

4. Define an algorithm that checks if two binary search trees are same.

   A and B below are same. A and B are not since the structures differ. A and D are not same since the keys differ.

```
  3           3           3           3
 / \         / \         / \         / \
2   5       2   5       2   5       2   5
   / \         / \   / \           / \
  1   4       1   4   1   4           4   1

  A           B           C           D
```

5. Define an algorithm that prints keys in level order. That is, the keys at level i are printed before the keys at level $i + 1$ are printed.

   What is the time and space complexity of the algorithm?

6. 2-3-tree is a type of balanced search tree.

   A tee is 2-3-tree if the following are true:

   - Every non leaf node has 2 or 3 childeren
   - All the leaf nodes have same depth

   (a) Draw the greatest and smallest 2-3-tree with height 1, 2 and 3

(b) Assume that 2-3-tree has height h. Show what is the smaller and upper limit of the number of the nodes.

(c) Assume that 2-3-tree has n nodes. Show what is the smaller and upper limit of the height of the tree.

The following formula might be useful: $\sum_{i=0}^{k} a^k = \frac{a^{k+1}-1}{k-1}$ where $a \neq 1$.