1. (recap)
   Give an algorithm in $O(n \log n)$ that takes as input a sequence $A$ of $n$ integers, and gives back as output a sequence in which every integer from $A$ occurs exactly once.

2. (recap)
The recurrence for merge sort is:

   $$T(n) = \begin{cases} 
   1 & \text{als } n = 1 \\
   2T\left(\frac{n}{2}\right) + n & \text{als } n > 1
   \end{cases}$$

   Solve the recurrence for $n$ a power of 2, and argue that the worst-case time complexity of merge sort is hence in $O(n \log n)$.

3. What is the shape of a tree with the highest possible level numbering using the minimum number of nodes? (We have seen this already.)

4. Give pseudo-code for the algorithm of level order traversal of a binary tree. (We visit the nodes layer by layer from top to bottom and per layer from left to right.)

5. Which node is visited last in a preorder, inorder, postorder traversal?

6. Give pseudo-code of a non-recursive algorithm for inorder traversal. Use a stack. Also describe step by step the result of applying your algorithm to the following tree:

   $\begin{array}{c}
   e \\
   / \ \ / \\
   b \ a \ c
   \end{array}$

7. Give pseudo-code for an algorithm that takes as input a node $v$ in a binary tree, and gives as output the node that is visited directly after $v$ in case of a preorder, inorder, postorder traversal.

8. Give pseudo-code of the algorithm $lca$ that takes as input two nodes $v$ and $w$ in a tree $T$, and that gives as output the least common ancestor of $v$ and $w$ in $T$.

   What is the time complexity of your algorithm in terms of $O$?
9. Consider the array representation of a heap with 15 elements. What is the sequence of indices for a preorder traversal of the heap? And for the inorder and postorder traversal?

10. Give an example showing that a preorder traversal of a min-heap does not necessarily yield a non-decreasing sequence.

11. Give an example showing that a preorder traversal of a min-heap does not necessarily yield a non-increasing sequence.

12. Depict binary trees with heights 2, 3, 4, 5, and 6, all with the seven keys 1, 4, 5, 10, 16, 17, 21.

13. Build a binary search tree by adding one by one, and in this order, the following numbers: 5, 6, 3, 8, 7, 4, 1, 2.

14. How many binary search trees with labels 1, 2, 3, 4 exist? Is it relevant what the labels are? And that it is a binary search tree?

15. For every of the following statements: explain shortly why it is true or give a counterexample showing it is false.
   
   (a) There exists a binary search tree which is also a min-heap.
   
   (b) There exists a binary search tree which is also a max-heap.
   
   (c) Every binary search tree is a min-heap.
   
   (d) Every min-heap is a binary search tree.

16. Assume a binary search tree with labels the numbers 1 till 1000. We search a node labeled 363. Which of the following sequences cannot be the labels of nodes visited in searching 363?

   (a) 2, 252, 401, 398, 330, 344, 397, 363.

   (b) 924, 220, 911, 244, 898, 258, 362, 363.

   (c) 925, 202, 911, 240, 912, 245, 363.

   (d) 2, 399, 387, 219, 266, 382, 381, 278, 363.

   (e) 935, 278, 347, 621, 299, 392, 358, 363.

17. Give pseudo-code for a recursive algorithm that finds the node with the smallest label in a sub-tree starting with node v. Same for finding the node with the largest key, starting with node n.

18. Add 4 to the following binary search tree, and next add 14:
19. Remove 6 from the following binary search tree, and next remove 14.

![Binary Search Tree Diagram]

20. We can sort a sequence as follows: add the numbers one by one to a binary search tree, then visit the nodes using inorder. What is the best-case and the worst-case time complexity of this algorithm?