Exercise 1

Write an algorithm that copies all keys that are stored in a binary search tree into an array of appropriate size. In the resulting array, the keys shall be sorted in descending order.

Exercise 2

Consider the binary tree given by the expression

\[ x = (5, (3, \text{emptyTree}, (4, \text{emptyTree}, \text{emptyTree})), \]
\[ (8, (6, \text{emptyTree}, \text{emptyTree}), (10, (9, \text{emptyTree}, \text{emptyTree}), \]
\[ (13, \text{emptyTree}, \text{emptyTree}))) \]

- draw a diagram of this binary tree and decide whether its a binary search tree
- perform the following operations (using the resp. algorithms from the lectures), and draw a diagram of the search tree after each operation:
  - TreeInsert\( (x,11) \)
  - TreeDelete\( (x,5) \)
  - TreeInsert\( (x,5) \)
  - TreeInsert\( (x,12) \)

Exercise 3

Decide whether the binary tree given in Exercise 2 is an AVL tree

- before the insert/delete operations, and
- after each of the regular insert/delete operations.

Again, perform the insert/delete operations given in exercise IV, and name and perform the rotation(s) to restore the AVL property after each step (if required). Draw a diagram of the search tree after each of your insert/delete, or rotation operations.