University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
G. Schmid

Algorithms and Datastructures
Winter Term 2022
Exercise Sheet 6

Due: Wednesday, December 06th, 2pm

Exercise 1: Binary Search Tree - Range Queries *(10 Points)*

(a) Implement the binary search tree (BST) data structure and the `insert` operation. You can use the template `BST.py`. *(4 Points)*

(b) Implement the operation `getrange(`x_{min}, x_{max}`)` efficiently on binary search trees which returns all keys x in the tree with $x_{min} \leq x < x_{max}$ (cf. lectue notes week 6 slide 21). *(4 Points)*

(c) Use your implementation of BST and your `insert` function to insert all words from the file `inputs.txt` into a BST with respect to the lexicographic ordering on words over the alphabet $\{a, \ldots, z\}$[1]. Use your data structure to output all words from the BST beginning with a certain prefix.[2] Output all words with prefix "qw". Copy the result into your `experiences.txt` file. *(2 Points)*

Exercise 2: Binary Search Tree - Operations *(10 Points)*

(a) Describe a function which returns the depth of a binary search tree and analyze the runtime. *(2 Points)*

(b) Describe a function that for a given binary search tree with n nodes and a given $k \leq n$ returns a list with the k smallest keys from the tree. Analyze the runtime in dependence of k and the depth of the tree d. *(4 Points)*

(c) Describe a function that takes a binary search tree B and a key x as input and generates the following output:

- If there is an element v in B with $v.key = x$, return v.
- Otherwise, return the pair (u, w) where u is the tree element with the next smaller key and w is the element with the next larger key. It should be $u =$ None if x is smaller than any key in the tree and $w =$ None if x is larger than any key in the tree.

For your description you can use pseudo code or a sufficiently detailed description in English. You can use the methods of the lecture as a black box.

Analyze the runtime of your function. *(4 Points)*

[1] Python supports the comparison of strings with respect to the lexicographic ordering, i.e., you can use "<", "<=".

[2] If you enter `Python3` and `from BST import BST` into the command prompt you can use the class `BST` from the command line. We provided a method for inserting the content of `inputs.txt`.