

5LN446/5LN715 Assignment 3

Six tasks: 5LN446 complete three tasks. 5LN715 complete five tasks.

Submission only need to include concisely commented and well-designed code, written reports are not required. Please make sure that your code complies with the specification, above all that all classes, functions and methods take the right number of arguments, and return objects of the right type. In well-designed code functions and variables are named in such a way that their point is easy to grasp. (For more information see course PM.) The tasks are to some extent open in character: Try to decide on the most reasonable interpretation in that case.

Files provided:

- `Tree.py`: General (syntax) tree. Tree and node class the same – no provision for empty trees.
- `BSTree.py`: Binary search tree. Separate tree and node classes – support empty trees.
- Plotting: `plot1.zip`. Or in the Linux system: `/home/staff/matsd/plt2/plot1` Library `matplotlib` required (if you run it on your own Python installation). Run in this way on the Linux system: `/local/new/bin/python3 sortPlot.py`. You'll get the first plot for free; the second one requires Tree Sort, which is left as an assignment.

(Assignment 3.1) General (syntax) tree class (October 10 slides, `Tree.py`) – design class methods: (a) retrieves the list of leaf values (i.e. the sequence of words in the syntax application shown in the slides); (b) finds the number of the largest number of subtrees (daughters) in a tree.

(Assignment 3.2) Binary search tree class (October 13 slides, `BSTree.py`) – design class methods: (a) gives the maximum depth of the tree; (b) inserts new values into the tree; (c) retrieves the sorted list of values; (d) the sorting procedure defined by combining (b) and (c) to be called as one non-class method (this style: `slist = treeSort(alist)`).

(Assignment 3.3) Before, we have used `BSTree` as a set of values. We can extend it to what is called a mapping associating keys with values. Imagine a dictionary associating words (keys) with part-of-speech labels (values). Implement this idea! (In real life you'd probably use Python's `dict` type.)

```
posmap = BSTreeMap()
posmap.put('duck', 'verb')
posmap.put('duck', 'noun')
print(posmap.get('duck')) # should return ['verb', 'noun']
```

(Assignment 3.4) Binary search tree class (October 13 slides, `BSTree.py`) – implement deletion (look for the algorithm in the literature/on the net).

(Assignment 3.5) Implement Merge Sort in a version that produces a tracing of what is happening (more or less in the style of the October 13 slides) and plots its performance compared to Bubble and Selection Sort (also in the style of the October 13 slides).

(Assignment 3.6) Implement some other sorting algorithm as required in the previous task (for Merge Sort). See: en.wikipedia.org/wiki/Sorting_algorithm for an overview.