Exercise sheet 1

February 18, 2020

Exercise 1 Insertion Sort *(10 points)*
Implement the *insertion sort* algorithm. The algorithm takes the first element of the unsorted list and inserts it in the correct place of the sorted list. All elements with a higher index than the inserted element have to be reallocated. For a graphical example look at `https://en.wikipedia.org/wiki/Insertion_sort`. Write at least one unit test for each function, for both expected input as well as unexpected input cases. As for the expected input example, you e.g. can test if your algorithm works with an even and an uneven number of elements. As for the unexpected input case, you could check how the function behaves in case of an empty input list.

Exercise 2 Binary Heap *(5 points)*
The following binary *min-heap* is stored in a linear array. Draw the heap as a tree diagram and mark the edges where the *heap condition* is not fulfilled.

$$[1, 6, 2, 4, 7, 3, 9, 11, 13, 12, 14, 5, 15, 8, 10, 13, 13]$$

Commit Subversion, Daphne, Jenkins *(5 points)*
Please register (if not done yet) for the course in *Daphne* provided under the following link:
`https://daphne.informatik.uni-freiburg.de/ws1819/AlgoDatEse/`
Pay attention to the correctness of your provided data, especially your e-mail address.
Commit your solutions via svn inside a new subfolder **uebungsblatt_01** into your SVN repository. Also commit the "Makefile" you find in the Python template on the course website. Do not commit temporary files, e.g. the ones created by Python (you can remove these via "*make clean*"). For the second exercise, commit your solution in PDF format in a subfolder named "*non-code*". Please also commit a text file *erfahrungen.txt* into the folder. In there

describe your experience with the exercise sheet in a few sentences: Was it manageable for you? How much time did it take you? Did you have problems with specific parts?