



Algorithms and Datastructures

Summer Term 2022

Exercise Sheet 1

Due: Wednesday, May 4th, 4 pm

Exercise 1: Registration

(5 Points)

Register for the online course system Daphne. You can also find the according link on the Website of this course. Make sure that your data is correct, specifically that you can be reached under the given email address. Then execute the *checkout* command on your SVN-repository.¹

Exercise 2: Quicksort

(5 Points)

Implement the algorithm *QuickSort* from the lecture with two different options of how to choose the pivot element: "Element at first position", "Element at random position". Use the template `QuickSort.py` that is provided on the website. Write a unit test for both the `quicksort_divide` and the `quicksort_recursive` method. The unit tests should check at least one non-trivial example. If there are critical cases that are easy to check (e.g., an empty input), you should make a unit test for these cases, too.

Exercise 3: Time Measurement

(5 Points)

Measure the runtime of your *QuickSort* implementation for the two variants of choosing the pivot and for two different kinds of inputs. The first kind of inputs are reversed arrays i.e. arrays of the form $[n, n - 1, \dots, 2, 1]$, the second kind are arrays filled with n random integers. Repeat this for input sizes $n \in \{100, 200, \dots, 5000\}$.² Plot the runtimes of all 4 variants (pivot, input) into the same chart.³ Use your plots to compare the runtimes and write a short evaluation into the file `experience.txt` (c.f., Task 4).

Exercise 4: Submission

(5 Points)

Commit your code including the tests and the plots into the SVN, into a subfolder `exercise-01` (German for exercise sheet 01). Make sure that there are no errors when you run your code (including style check and unit tests) on Jenkins. Commit a file `experience.txt` in which you describe your experiences with this exercise sheet and any problems that may have appeared.

¹Your SVN-repository will be created automatically after your registration to Daphne. The URL is <https://daphne.informatik.uni-freiburg.de/ss2022/AlgoDatCond/svn/your-rz-account-name>

²A function to generate the arrays and the time measurements is provided in `QuickSort.py`

³The differences in runtimes will be most distinct if they are plotted in a single chart with n on the x -axis and the runtime $T(n)$ on a *linear* and *logarithmic* y -axis.