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

Algorithms and Datastructures
Summer Term 2024
Exercise Sheet 1

Due: Wednesday, April 24th, 12 pm

Exercise 1: Registration *(5 Points)*

Register for Zulip using the invitation-link given on the website. Note that we use Zulip as **forum** for questions regarding the *lecture* and the *exercises* as well as the platform to hand in submissions to tutors.

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 $[n, n-1, \ldots, 2, 1]$, the second kind are arrays filled with n random integers.
Repeat this for input sizes $n \in \{100, 200, \ldots, 5000\}$.[1] Plot the runtimes of all 4 variants (pivot, input) into the same chart.[2] 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)*

Zip your code including the tests and the plots together in one file and send them to your tutor.

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

[2] The 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.