

Algorithms and Datastructures Exercise Sheet 1

Exercise 1: Quicksort

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 2: Time Measurement

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, ..., 2, 1], the second kind are arrays filled with n random integers.

Repeat this for input sizes $n \in \{100, 200, \dots, 5000\}$.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).

(10 Points)

(10 Points)

 $^{^{1}\}mathrm{A}$ function to generate the arrays and the time measurements is provided in QuickSort.py

²The 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.