

# Hausübung 8 - Algorithmen

Jonathan von Rad  
Benedikt Schaber

December 15, 2022

## Aufgabe 1

siehe Code: [Github Repository](#)

## Aufgabe 2

Das Pivot Element spielt eine große Rolle → Es fungiert als **Splitter**. Alle Elemente in dem (Sub)Array, werden mit diesem verglichen und **relativ zu diesem sortiert**. Also wenn das Pivot Element z.B. 10 ist, werden alle Zahlen kleiner als 10 auf die eine Seite und alle größer als 10, auf die andere Seite geplaced. Ideal ist hierbei ein Pivot-Element, welches das Array in zwei gleich große Partitions teilt → der **Median**.

In unserer Implementierung wird immer das letzte Element des Arrays als Pivot Element gewählt.

## Aufgabe 3

- Selection Sort:  
Worst Case existiert nicht.  
→  $T(n) = \Theta(n^2)$
- QuickSort:  
Worst Case = Reverse Sortierte Liste, da immer das rechteste Element der Splitter ist und dann die SubArrays nur (n-1), (n-2),... groß werden  
→  $T(n) = O(n)$

## Aufgabe 4

Für Arrays siehe Code: [Github Repository](#)

Wir haben die Testfälle so gewählt, das alle Randfälle ausgetestet werden:

- Reversed Sorted List
- Sorted List
- Negative Zahlen
- Exponentiell wachsende Zahlen

## Aufgabe 5

siehe Code: [Github Repository](#)

## Aufgabe 6

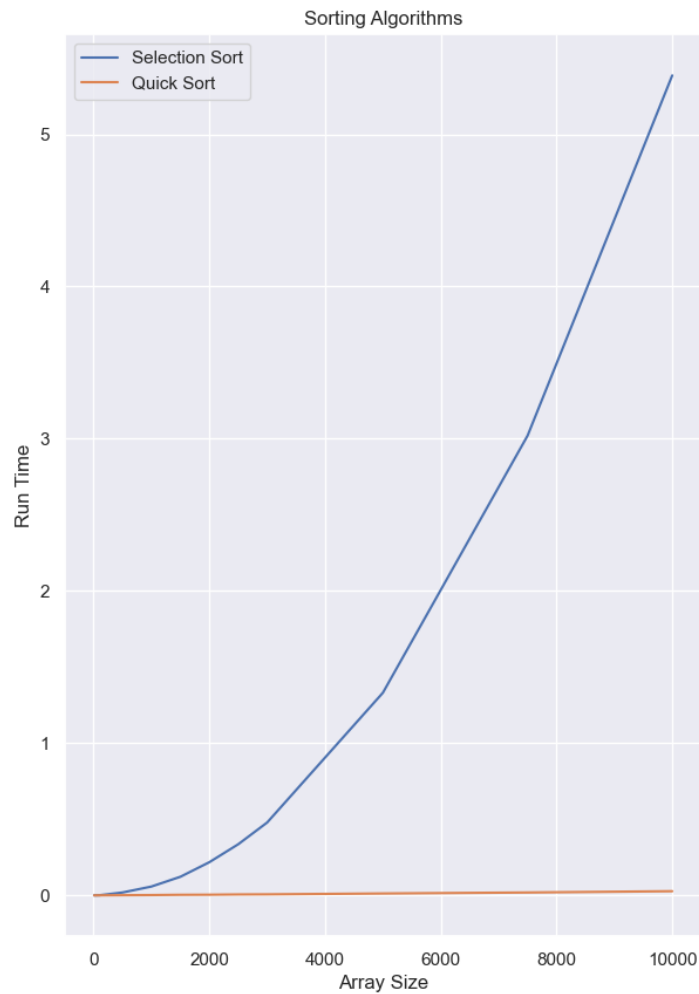


Figure 1: This is a plot showing the relation between input-size and run-time of Selection Sort and Quick Sort.

Ab Input Größen von 500 Elementen, ist ein signifikanter Unterschied bemerkbar (10x time).

## Aufgabe 7

siehe Code: [Github Repository](#)

Konstante von Selection Sort: 5.386692047119141e-08

Konstante von Quick Sort: 2.97013011926992e-07

## Aufgabe 8 - 10

siehe Code: [Github Repository](#)