Sortieralgorithmen **Prüfungsstoff**

1. Du kannst die folgenden Sortieralgorithmen für einfache Listen tabellarisch durchführen.

   - Insertionsort
   - Selectionsort
   - Bubblesort
   - Quicksort (mit Pivot-Index $m = \lfloor n/2 \rfloor$)

2. Du kannst für die oben genannten Sortieralgorithmen die Anzahl der Vergleiche und der Vertauschungen bestimmen.

3. Du kannst die oben genannten Sortieralgorithmen an ihrem Python-Programmcode erkennen.

4. Du kannst die mittlere Laufzeitkomplexität der oben genannten Algorithmen angeben und verstehst damit, wie stark die Laufzeit anwächst, wenn die Eingabegrösse (die Anzahl der zu sortierenden Elemente) z. B. verdoppelt oder verzehnfacht wird.

5. Du kannst die folgenden zwei Pivot-Strategien beschreiben:

   - randomized
   - Median-of-three

6. Du kannst Situationen erkennen (Best-Case, Worst-Case), in denen die Sortieralgorithmen eine bessere bzw. schlechtere Laufzeitkomplexität haben. Beispielsweise hat Insertionsort bei einer bereits sortierten Liste eine Laufzeitkomplexität von $\mathcal{O}(n)$