

1. Du kannst die folgenden Sortieralgorithmen für einfache Listen tabellarisch durchführen.
  - Selectionsort
  - Insertionsort
  - Bubblesort
  - Quicksort
2. Du kannst für Selectionsort und Bubblesort die Anzahl der Vergleiche und die Anzahl der Vertauschungen bestimmen sowie für Insertionsort die Anzahl der Vergleiche und Speicheroperationen (Verschiebungen) angeben.
3. Du kannst den Partitionierungsschritt von Quicksort auf einer Teilliste tabellarisch durchführen.
4. Du kannst geeignete Arrays mit Quicksort sortieren.
5. Du kannst die oben genannten Sortieralgorithmen an ihrem Python-Programmcode erkennen.
6. Du kannst die oben genannten Sortieralgorithmen anhand eines geeigneten Sortierprotokolls erkennen.
7. Du kannst angeben, wie schnell die von uns im Unterricht behandelten Sortieralgorithmen
  - Selectionsort
  - Insertionsort
  - Bubblesort
  - Quicksortim Best Case und im Worst Case sortieren, indem du die zur Arraylänge proportionale ungefähre Sortierdauer angibst.
8. Du kannst quantitativ bestimmen, wie sich die Sortierdauer verändert, wenn die Arraygröße vergrößert oder verkleinert wird.
9. Du kannst beschreiben, für welche Art von Listen Quicksort eine Worst Case-Laufzeit aufweist.
10. Du kannst zwei Massnahmen beschreiben, mit denen sich die Worst Case-Laufzeit von Quicksort vermeiden lässt.
11. Du kannst den Median einer Liste von Zahlen bestimmen.
12. Du kannst eine Python-Funktion schreiben, die überprüft, ob eine Liste bereits sortiert ist und dann `True` zurückgibt und sonst `False`.
13. Du kannst eine Python-Funktion schreiben, die den kleinsten Wert in einer Liste bestimmt und als Wert zurückgibt. (Das wird bei Selectionsort verwendet.)