

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