

- Du weisst, dass die Laufzeitkomplexität ein Hilfsmittel ist, das die Komplexität eines Algorithmus' unabhängig von der Hardware und der Programmiersprache als Funktion der Anzahl  $n$  der Eingabedaten darstellt.
- Du verstehst die Big-Oh-Notation und kannst solche Ausdrücke vereinfachen.
- Du kennst von folgenden Algorithmen die Laufzeitkomplexität:
  - Einen Wert in eine Liste schreiben, bzw. aus einer Liste lesen:  $O(1)$
  - Einen Wert in einer sortierten Liste suchen:  $O(\log_2 n)$
  - Einen Wert in einer unsortierten Liste suchen:  $O(n)$
  - Fortgeschritten Sortieralgorithmen (Quicksort):  $O(n \cdot \log_2 n)$
  - „Naive“ Sortieralgorithmen (Insertionsort, Selectionsort, Bubblesort):  $O(n^2)$
  - Travelling Salesman Problem:  $O(n!)$
- Du kannst den Zeitbedarf von Algorithmen mit bekanntem  $O$  abschätzen, wenn man die Eingabegrösse  $n$  verändert.
- Du kannst beschreiben, welche Laufzeitkomplexitäten für grosse  $n$  praktisch nicht mehr berechenbar sind.