

1. Du kannst Gnomesort schrittweise ausführen, indem du die Position des Gartenzwerchs jeweils angibst.
2. Du kannst Selectionsort schrittweise ausführen und dabei die Anzahl der Vergleiche und die Anzahl der Vertauschungen bestimmen.
3. Du kannst Insertionsort schrittweise ausführen und dabei die Anzahl der Vergleiche und der Verschiebeoperationen angeben.
4. Du kannst die iterartive Version von Mergesort schrittweise ausführen.
5. Du verstehst die Merge-Funktion soweit, dass du bis maximal 4 Zeilen im Programmcode ergänzen kannst.
6. Du kennst einen zentralen Nachteil von Mergesort gegenüber den anderen betrachteten Sortierverfahren.

Mergesort muss die zu sortierenden Teillisten zwischenspeichern, und hat eine Speicherkomplexität von $O(n)$. Die anderen Sortierverfahren (Gnomesort, Selectionsort, Insertionsort) benötigen maximal ein Speicherplatz beim Vertauschen oder Verschieben von Elementen, was $O(1)$ bedeutet.

7. Du kannst die oben genannten Sortieralgorithmen an ihrem Python-Programmcode erkennen.
8. Du kannst die oben genannten Sortieralgorithmen anhand eines geeigneten Sortierprotokolls erkennen.
9. Du kannst angeben, wie schnell die von uns im Unterricht behandelten Sortieralgorithmen
 - Gnomesort
 - Selectionsort
 - Insertionsort
 - Mergesort

im Best Case und im Worst Case sortieren, indem du die zur Arraylänge proportionale ungefähre Sortierdauer angibst.

10. Du kannst quantitativ bestimmen, wie sich die Sortierdauer verändert, wenn die Arraygröße vergrößert oder verkleinert wird.