

20. Überlegungen zur **Komplexität** von Algorithmen:

Für den Aufwand  $A(n)$  und folglich den Zeitbedarf zur Laufzeit des Algorithmus gilt bei

- SelectionSort:  $A(n) \sim n^2$
- MergeSort:  $A(n) \sim n \cdot \log_2(n)$
- Fibonacci-Folge:  $A(n) \sim 2^n$  (bei rekursiver Berechnung)
- BinarySearch:  $A(n) \sim \log_2(n)$

Entsprechend haben

- SelectionSort quadratische Komplexität,
- MergeSort linear-logarithmische Komplexität,
- die rekursive Berechnung der Fibonacci-Folge exponentielle Komplexität,
- BinarySearch logarithmische Komplexität.

Vergleiche die Algorithmen SelectionSort, MergeSort und BinarySearch hinsichtlich ihrer Effizienz zur Laufzeit, indem man jeweils eine Anzahl  $n$  der zu verarbeitenden Datenelemente auf das 1000-fache erhöht.

Wähle  $n = 100$  und bilde jeweils den Quotienten  $A(1000 \cdot n)/A(n)$ .