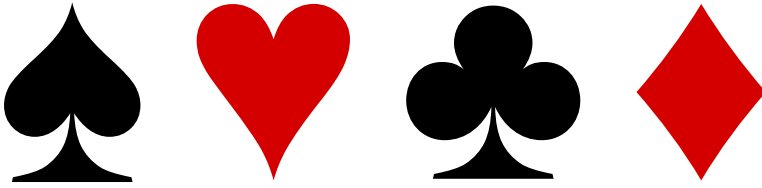


103 Algorithmen und dynamische Datenstrukturen

Aufgabe: Sortialgorithmen verstehen und anwenden



Für einen der unten aufgelisteten Sortialgorithmen sollen die Eigenschaften stabil/instabil, iterativ/rekursiv, in-place/out-of-place geklärt und die Zeitkomplexität (O-Notation) im Best-, Average-, Worst-Case ermitteln werden.

Außerdem soll jeder Algorithmus in Form eines Struktogramms (Nassi-Shneiderman-Diagramm) oder eines Programmablaufplans (PAP, engl. flowchart) graphisch dargestellt und anschließend in Go implementiert werden. Die implementierten Funktionen sollen ein Slice von Ganzzahlen beliebiger Menge sortieren können. Die Sortierfunktion soll die nachfolgende Signatur haben:

```
func Sort(a []int) []int {  
    // TODO: implement  
}
```

Abschließend sollen die Implementierungen vorgestellt und der Ablauf der Algorithmen anhand von Spielkarten demonstriert werden.

- ☐ Heapsort
- ☐ Insertionsort
- ☐ Mergesort
- ☐ Quicksort
- ☐ Selectionsort
- ☐ Shellsort

Links

- ↗ Sortialgorithmen - Bleeptrack
(<https://deprecated.bleeptrack.de/tutorials/sortialgorithmen/>)
- ↗ Algorithmentänze - YouTube (<https://www.youtube.com/user/AlgoRythmics>)
- ↗ Heapsort - Wikipedia (<https://de.wikipedia.org/wiki/Heapsort>)
- ↗ Insertionsort - Wikipedia (<https://de.wikipedia.org/wiki/Insertionsort>)
- ↗ Mergesort - Wikipedia (<https://de.wikipedia.org/wiki/Mergesort>)
- ↗ Quicksort - Wikipedia (<https://de.wikipedia.org/wiki/Quicksort>)
- ↗ Selectionsort - Wikipedia (<https://de.wikipedia.org/wiki/Selectionsort>)
- ↗ Shellsort - Wikipedia (<https://de.wikipedia.org/wiki/Shellsort>)