Gruppe B
Selection-Sort

Vorgehen:
Erarbeiten Sie zusammen das Verfahren „Selection-Sort“. Lesen Sie dazu den kurzen Einführungstext auf dieser Seite und teilen Sie die übrigen Teile untereinander auf:

1. Textbeschreibung
2. Struktogrammdarstellung
3. Pseudo-Code

Erklären Sie sich das Verfahren gegenseitig. Überlegen Sie, wie Sie es den anderen Schüler:innen erklären können. Entwerfen Sie eine Präsentation und ein Handout zu dem Verfahren, das Sie den übrigen Schüler:innen zur Verfügung stellen können. Jedes Mitglied der Gruppe muss in der Lage sein, das Verfahren zu erklären.

Begründen Sie auch, warum das Verfahren eine Sortierung zur Folge hat. Zählen Sie die Vergleiche, die ausgeführt werden für:

a = [1, 2, 3, 4, 5, 6]
b = [6, 5, 4, 3, 2, 1]

Entscheiden Sie, ob dieses Sortierverfahren stabil ist. Begründen Sie Ihre Entscheidung.

Teilung des Arrays in einen sortierten und einen unsortierten Teil

Bei den Sortieralgorithmen, die wir besprechen, wird das Array in einen sortierten und einen unsortierten Teil geteilt. In Ihrer Präsentation sollte immer deutlich werden, welcher Teil sortiert und welcher Teil unsortiert ist.
Um es einheitlicher zu gestalten, sollten Sie den sortierten Teil mit grüner und den unsortierten Teil mit roter Farbe verdeutlichen.

Ausgangspunkt der Sortierung ist folgendes unsortiertes Array

array = [18, 10, 38, 11, 23, 24, 45, 11]

In der grafischen Darstellung:

18	10	38	11	23	24	45	11

Selection-Sort: Textbeschreibung

Bei Selection-Sort wird für jeden Index des Arrays das passende Element ausgesucht. Man fängt mit dem Index 0 an und sucht den Index des Minimums der unsortierten Liste.

↓	min						
18	10	38	11	23	24	45	11

Dann werden die Elemente des aktuellen Indexes mit dem des Minimums getauscht. Nun geht man zum nächsten Index und sucht im unsortierten Teil der Liste nach dem Minimum. Da 11 zweimal in der Liste vorkommt, wird der Index des ersten Vorkommens verwendet. Dann wird das Element mit dem Index 1 mit dem Element mit dem Index des Minimus getauscht.

	↓		min				
10	18	38	11	23	24	45	11

		↓					
10	11	38	18	23	24	45	11

Im weiteren Fortgang wird der Index 2 untersucht. Das Minimum der unsortierten Liste ist die zweite 11, deswegen werden 38 und die 11 (das Element mit dem Index 7) getauscht.

			↓				
10	11	11	18	23	24	45	38

Das Minimum der unsortierten Liste ist in den folgenden drei Fällen jeweils der Index, der untersucht wird.

				↓			
10	11	11	18	23	24	45	38

					↓		
10	11	11	18	23	24	45	38

						↓	
10	11	11	18	23	24	45	38

Beim vorletzten Index findet noch einmal ein Tausch statt.

							↓
10	11	11	18	23	24	38	45

10	11	11	18	23	24	38	45

Bei der Untersuchung des letzten Indexes hat man den Fall einer einelementigen Liste. Diese ist immer sortiert. Die Liste ist sortiert.

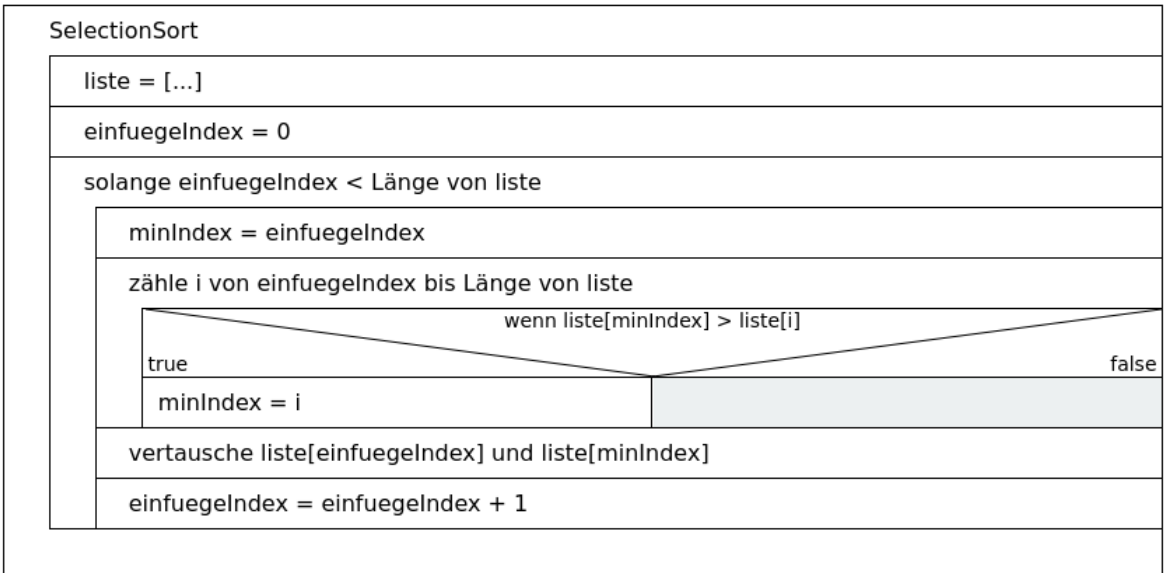
Selection-Sort: Pseudo-Code

```
liste = […]
einfuegeIndex = 0
solange einfuegeIndex < len(liste)
     minIndex = einfuegeIndex
     für i von einfuegeIndex … Länge der Liste
          wenn liste[minIndex] > liste[i]
               minIndex = i
     vertausche liste[einfuegeIndex] und liste[minIndex]
     einfuegeIndex = einfuegeIndex + 1
```

Übertragen Sie diesen Pseudo-Code in eine Python-Funktion selectionSort(liste) und testen Sie ihn.

Spielen Sie den Code mit den Karten durch.

Selection-Sort: Struktogramm



Führen Sie mit einem kurzen Array:
test = [20, 10, 30, 5, 10]
einen Trockentest durch.