

Gruppe C

Bubble-Sort

Vorgehen:

Erarbeiten Sie zusammen das Verfahren „Bubble-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
----	----	----	----	----	----	----	----

Bubble-Sort: Textbeschreibung

Bei Bubble-Sort wird das Array mehrmals durchgegangen. Bei jedem Durchgang werden immer benachbarte Elemente miteinander verglichen. Ist das linke Element größer als das rechte, werden die Elemente getauscht.

Hier eine schrittweise Darstellung des ersten Durchgangs:

18	10	38	11	23	24	45	11
----	----	----	----	----	----	----	----

Da $\text{Array}[0] > \text{Array}[1]$: Tausch der Elemente

10	18	38	11	23	24	45	11
----	----	----	----	----	----	----	----

$\text{Array}[1] < \text{Array}[2]$: kein Tausch

10	18	38	11	23	24	45	11
----	----	----	----	----	----	----	----

$\text{Array}[2] > \text{Array}[3]$: Tausch

10	18	11	38	23	24	45	11
----	----	----	----	----	----	----	----

$\text{Array}[3] > \text{Array}[4]$: Tausch

10	18	11	23	38	24	45	11
----	----	----	----	----	----	----	----

$\text{Array}[4] > \text{Array}[5]$: Tausch

10	18	11	23	24	38	45	11
----	----	----	----	----	----	----	----

$\text{Array}[5] < \text{Array}[6]$: kein Tausch

10	18	11	23	24	38	45	11
----	----	----	----	----	----	----	----

$\text{Array}[6] < \text{Array}[7]$: Tausch

10	18	11	23	24	38	11	45
----	----	----	----	----	----	----	----

Da wir am Ende des Arrays angekommen sind, ist das Maximum so nach oben gestiegen. Deswegen baut sich vom Arrayende her die sortierte Liste auf.

In den nächsten Durchläufen steigt jeweils das Maximum der unsortierten Liste nach oben.

Array nach dem 2. Durchlauf:

10	11	18	23	24	11	38	45
----	----	----	----	----	----	----	----

Array nach dem 3. Durchlauf:

10	11	18	23	11	24	38	45
----	----	----	----	----	----	----	----

Array nach dem 4. Durchlauf:

10	11	18	11	23	24	38	45
----	----	----	----	----	----	----	----

Array nach dem 5. Durchlauf:

10	11	11	18	23	24	38	45
----	----	----	----	----	----	----	----

In den folgenden Durchläufen gibt es in dem Array jetzt keine Veränderungen mehr.

10	11	11	18	23	24	38	45
----	----	----	----	----	----	----	----

10	11	11	18	23	24	38	45
----	----	----	----	----	----	----	----

10	11	11	18	23	24	38	45
----	----	----	----	----	----	----	----

10	11	11	18	23	24	38	45
----	----	----	----	----	----	----	----

Nach allen Durchläufen ist die Liste sortiert.

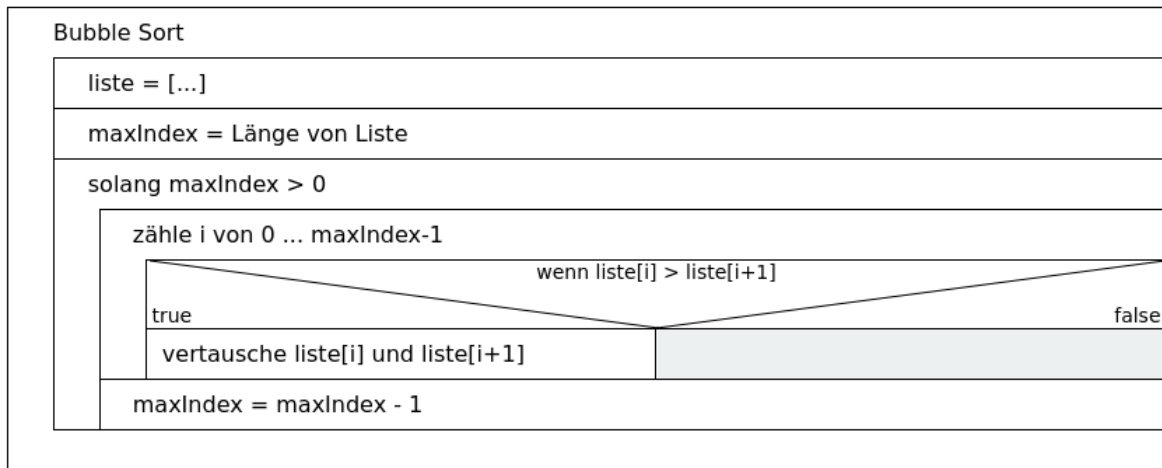
Bubble-Sort: Pseudo-Code

```
liste = [...]  
maxIndex = Länge von liste  
solange maxIndex > 0  
    für i von 0 ... maxIndex-1  
        wenn liste[i] > liste[i+1]  
            vertausche liste[i] und liste[i+1]  
    maxIndex = maxIndex-1
```

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

Spielen Sie den Code mit den Karten durch.

Bubble-Sort: Struktogramm



Führen Sie mit einem kurzen Array:

`test = [20, 10, 30, 5, 10]`

einen Trockentest durch.