2.5 Algorithmen im Alltag - Sortierverfahren

Einstieg:

- Im vorliegenden Zitat werden Algorithmen als Helfer im Alltag beschrieben, die diesen in vielen Fällen sogar beeinflussen, wobei dies meist unbewusst geschieht.
- Es sind individuelle Lösungen möglich.

  Beispiele: Navigationsalgorithmen, Sortieralgorithmen, Kochrezept, Suchalgorithmen (Suchmaschinen, z.B. Google), Empfehlungsalgorithmen (Online-Shops, Streaming-Dienste, etc.), Bilderkennung (z.B. Social Media), Kreditbewertung, Medizinische Diagnose, Automatische Spracherkennung (Siri, Alexa, …), Verschlüsselungsalgorithmen, Grafikalgorithmen (Videospiele), Verkehrssteuerungssysteme (z.B. Ampelschaltung)

A1

Es sind individuelle Lösungen möglich.

A2

a) Beim Bubblesort-Algorithmus wird die Liste immer wieder durchlaufen – so lange bis diese vollständig sortiert ist. Dabei werden benachbarte Elemente ständig auf die korrekte Reihenfolge verglichen und gegebenenfalls miteinander getauscht.

b)

```
wiederhole Anzahl-der-Elemente-der-Liste mal {
     wiederhole Anzahl-der-Elemente-der-Liste -1 mal {
          falls (Listenelement-an-aktuellerStelle > Lis-
tenelement-an-aktuellerStelle+1) {
               tausche diese beiden Elemente
          }
     }
}
```

© C.C.Buchner Verlag, Bamberg

c) Ursprungszustand: K, U, 7, 10, O, A, 9, 8

K, U, 7, 10, O, A, 9, 8	Tausch
U, K, 7, 10, O, A, 9, 8	Tausch
U, 7, K, 10, O, A, 9, 8	Tausch
U, 7, 10, K, O, A, 9, 8	Tausch
U, 7, 10, O, K, A, 9, 8	kein Tausch
U, 7, 10, O, K, A, 9, 8	Tausch
U, 7, 10, O, K, 9, A, 8	Tausch
U, 7, 10, O, K, 9, 8, A	Ende des ersten Durchlaufs der Liste
U, 7, 10, O, K, 9, 8, A	Tausch
7, U, 10, O, K, 9, 8, A	Tausch
7, 10, U, O, K, 9, 8, A	kein Tausch
7, 10, U, O, K, 9, 8, A	kein Tausch
7, 10, U, O, K, 9, 8, A	Tausch
7, 10, U, O, 9, K, 8, A	Tausch
…	
7, 10, U, O, 9, 8, K, A	Ende des zweiten Durchlaufs der Liste
7, 10, U, O, 9, 8, K, A	kein Tausch
7, 10, U, O, 9, 8, K, A	kein Tausch
7, 10, U, O, 9, 8, K, A	kein Tausch
7, 10, U, O, 9, 8, K, A	Tausch
7, 10, U, 9, O, 8, K, A	Tausch
…	
7, 10, U, 9, 8, O, K, A	Ende des dritten Durchlaufs der Liste
7, 10, U, 9, 8, O, K, A	kein Tausch
7, 10, U, 9, 8, O, K, A	kein Tausch
7, 10, U, 9, 8, O, K, A	Tausch
7, 10, 9, U, 8, O, K, A	Tausch
…	
7, 10, 9, 8, U, O, K, A	Ende des vierten Durchlaufs der Liste
7, 10, 9, 8, U, O, K, A	kein Tausch
7, 10, 9, 8, U, O, K, A	Tausch
7, 9, 10, 8, U, O, K, A	Tausch
…	
7, 9, 8, 10, U, O, K, A	Ende des fünften Durchlaufs der Liste
7, 9, 8, 10, U, O, K, A	kein Tausch
7, 9, 8, 10, U, O, K, A	Tausch
…	
7, 8, 9, 10, U, O, K, A	Ende des sechsten Durchlaufs der Liste
…	

© C.C.Buchner Verlag, Bamberg

d) Die Aussage trifft nur für den elementaren Bubblesort-Algorithmus zu. Es ist jedoch möglich, eine verbesserte Version zu implementieren. Diese beinhaltet zwei Verbesserungen:

- Der Algorithmus „merkt“, wenn in einem Listendurchlauf keine Elemente mehr vertauscht worden sind, dass die Liste bereits vollständig sortiert ist und terminiert.
- Der Algorithmus betrachtet bei jedem Listendurchlauf ein Element weniger, da nach jedem Durchlauf ein weiteres Element am Ende der Liste bereits richtig sortiert ist.

Wird der Bubblesort-Algorithmus mit diesen Verbesserungen implementiert, so benötigt er möglichweise weniger Durchläufe als Elemente zu sortieren sind.

Aufgabe 1

a) Ursprungszustand der Liste: 42, 27, 72, 94, 73, 98, 72, 28, 48, 84

Zustand nach 1.Durchlauf der Liste: 27, 42, 72, 73, 94, 72, 28, 48, 84, 98
Zustand nach 2.Durchlauf der Liste: 27, 42, 72, 73, 72, 28, 48, 84, 94, 98
Zustand nach 3.Durchlauf der Liste: 27, 42, 72, 72, 28, 48, 73, 84, 94, 98
Zustand nach 4.Durchlauf der Liste: 27, 42, 72, 28, 48, 72, 73, 84, 94, 98
Zustand nach 5.Durchlauf der Liste: 27, 42, 28, 48, 72, 72, 73, 84, 94, 98
Zustand nach 6.Durchlauf der Liste: 27, 28, 42, 48, 72, 72, 73, 84, 94, 98

b) siehe Snap!-Datei `2-5_Aufgabe1_Bubblesort_Loesung.xml`

Aufgabe 2

a) Der Quicksort-Algorithmus basiert auf dem Prinzip „Teile und Herrsche“. Eine Liste wird mithilfe eines sogenannten Pivot-Elements schrittweise immer weiter in Teillisten (die alle kleineren/größeren Elemente bzgl. des Pivot-Elements zusammenfassen) zerlegt.

© C.C.Buchner Verlag, Bamberg

b) Anmerkung: Als Pivot-Element wird hier stets das erste Element der (Teil-) Liste verwendet.

Ursprungszustand: K, U, 7, 10, O, A, 9, 8

K, U, 7, 10, O, A, 9, 8	Pivotelement
U, 7, 10, O, 9, 8, K, A	Bildung der Teil-Listen
U, 7, 10, O, 9, 8, K, A	neues Pivotelement in jeder Teil-Liste; einelementige Teilliste ist bereits vollständig sortiert
7, 10, 9, 8, U, O, K, A	Bildung der Teil-Listen
7, 10, 9, 8, U, O, K, A	neues Pivotelement in jeder Teil-Liste; einelementige Teilliste ist bereits vollständig sortiert
7, 10, 9, 8, U, O, K, A	Bildung der Teil-Listen
7, 10, 9, 8, U, O, K, A	neues Pivotelement in jeder Teil-Liste
7, 9, 8, 10, U, O, K, A	Bildung der Teil-Listen
7, 9, 8, 10, U, O, K, A	neues Pivotelement in jeder Teil-Liste
7, 8, 9, 10, U, O, K, A	Bildung der Teil-Listen
	Liste ist vollständig sortiert

© C.C.Buchner Verlag, Bamberg