

Algorithmen und Datenstrukturen
SS 2019

Übungsblatt 4: Sortieren

Aufgabe 4-1 *Einfaches Sortieren*

Laut Wikipedia sind die 10 höchsten Gebäude die folgend aufgelisteten Bauwerke:

Rang	Gebäude	Stadt	Höhe	Etagen	Baujahr
1	Burj Khalifa	Dubai (VAE)	828 m	163	2010
2	Shanghai Tower	Shanghai (VR China)	632 m	128	2015
3	Makkah Royal Clock Tower	Mekka (Saudi-Arabien)	601 m	120	2012
4	Ping An Finance Center	Shenzhen (VR China)	599 m	115	2016
5	Lotte World Tower	Seoul (Südkorea)	555 m	123	2017
6	One World Trade Center	New York City (USA)	541 m	94	2014
7	Guangzhou CTF Finance Center	Guangzhou (VR China)	530 m	111	2016
8	Tianjin CTF Finance Center	Tianjin (VR China)	530 m	97	2020
9	China Zun Tower	Peking (VR China)	528 m	108	2018
10	Taipei 101	Taipei (Taiwan)	508 m	101	2004

Sortieren Sie die sechs höchsten Gebäude nach der Anzahl der nutzbaren Etagen. Zählen Sie alle Vertauschungs- und Vergleichsoperationen und vergleichen Sie mit den theoretischen Komplexitätsschranken. Sie können zum Zählen auch ein (Python-) Programm nutzen. Diskutieren Sie die Stabilität der Algorithmen. Nutzen Sie jeweils die folgenden Sortieralgorithmen:

- (a) BubbleSort
- (b) SelectionSort
- (c) InsertionSort

Lösungsvorschlag:

Etagen: 163, 128, 120, 115, 123, 94

Bubble Sort:

[163, 128, 120, 115, 123, 94]	Vergleich von:	Swap
[128, 163, 120, 115, 123, 94]	128 163	1
[128, 120, 163, 115, 123, 94]	120 163	2
[128, 120, 115, 163, 123, 94]	115 163	3
[128, 120, 115, 123, 163, 94]	123 163	4
[128, 120, 115, 123, 94, 163]	94 163	5
[120, 128, 115, 123, 94, 163]	120 128	6
[120, 115, 128, 123, 94, 163]	115 128	7
[120, 115, 123, 128, 94, 163]	123 128	8
[120, 115, 123, 94, 128, 163]	94 128	9
[115, 120, 123, 94, 128, 163]	115 120	10
[115, 120, 123, 94, 128, 163]	120 123	10
[115, 120, 94, 123, 128, 163]	94 123	11
[115, 120, 94, 123, 128, 163]	115 120	11
[115, 94, 120, 123, 128, 163]	94 120	12
[94, 115, 120, 123, 128, 163]	94 115	13

Vergleiche: 15 Swaps: 13

Laufzeit siehe Vorlesungsfolien.

Bubble Sort ist stabil

Lösungsvorschlag:

Selection Sort:

[163, 128, 120, 115, 123, 94]	Vergleich von:	Swap
[163, 128, 120, 115, 123, 94]	128 163	
[163, 128, 120, 115, 123, 94]	120 128	
[163, 128, 120, 115, 123, 94]	115 120	
[163, 128, 120, 115, 123, 94]	123 115	
[163, 128, 120, 115, 123, 94]	94 115	94 163
[94, 128, 120, 115, 123, 163]	120 128	
[94, 128, 120, 115, 123, 163]	115 120	
[94, 128, 120, 115, 123, 163]	123 115	
[94, 128, 120, 115, 123, 163]	163 115	115 128
[94, 115, 120, 128, 123, 163]	128 120	
[94, 115, 120, 128, 123, 163]	123 120	
[94, 115, 120, 128, 123, 163]	163 120	
[94, 115, 120, 128, 123, 163]	123 128	
[94, 115, 120, 128, 123, 163]	163 123	123 128
[94, 115, 120, 123, 128, 163]	163 128	

Vergleichsoperationen: 15 Swaps: 3

Laufzeit siehe Vorlesungsfolien

Selection Sort ist instabil.

Lösungsvorschlag:

Insertion Sort:

[163, 128, 120, 115, 123, 94]	Vergleich von:	Swap
[128, 163, 120, 115, 123, 94]	163 128	1
[128, 120, 163, 115, 123, 94]	163 120	2
[120, 128, 163, 115, 123, 94]	128 120	3
[120, 128, 115, 163, 123, 94]	163 115	4
[120, 115, 128, 163, 123, 94]	128 115	5
[115, 120, 128, 163, 123, 94]	120 115	6
[115, 120, 128, 123, 163, 94]	163 123	7
[115, 120, 123, 128, 163, 94]	128 123	8
[115, 120, 123, 128, 163, 94]	123 120	-
[115, 120, 123, 128, 94, 163]	163 94	9
[115, 120, 123, 94, 128, 163]	128 94	10
[115, 120, 94, 123, 128, 163]	123 94	11
[115, 94, 120, 123, 128, 163]	120 94	12
[94, 115, 120, 123, 128, 163]	115 94	13

Vergleichsoperationen: 14 Swaps: 13

Laufzeit siehe Vorlesungsfolien

Insertion Sort ist stabil

Aufgabe 4-2 *WorstCase beim Sortieren*

Geben Sie jeweils eine 6-elementige Liste von ganzen Schlüsselwerten an, die das Worst-Case-Verhalten bei BubbleSort, SelectionSort sowie InsertionSort auslösen.

Lösungsvorschlag:

1. BubbleSort

Bei BubbleSort wird die Worst-Case-Laufzeit durch das Sortieren einer Liste, die in umgekehrter Richtung sortiert wird, erreicht. In diesem Fall wird jedes Element j mal durchgetauscht, wobei es i Elemente gibt. (i und j sind dabei die Iteratoren der jeweiligen for-Schleifen, analog zur Beispielimplementierung im Skript/den Folien.

Ein Beispiel wäre also die Liste 6, 5, 4, 3, 2, 1, falls man diese aufsteigend Sortieren will.

2. SelectionSort

SelectionSort ist kein adaptiver Sortieralgorithmus. Es tut also nichts zur Sache, wie ein Array aussieht; man benötigt immer die selbe Anzahl an Vergleichsoperationen. Die Anzahl der Swaps können verschieden sein, worst-case-Beispiel: 6, 5, 4, 2, 1, 3

3. InsertionSort

Wie beim BubbleSort wird auch hier das Worst-Case-Verhalten beim sortieren einer umgedrehten Liste erzeugt. Man steigt aus der inneren for-Schleife des Algorithmus aus, wenn dessen Index-Element größer als das der äußeren for-Schleife ist. Bei einer umgedreht sortierten Liste führt man die innere for-Schleife damit immer aus.

Als Beispiel wieder analog: 6, 5, 4, 3, 2, 1

