

Algorithmen und Komplexität

Zusammenfassung

Fabian Sigmund

11.07.2023

1 Sortieralgorithmen (Einfach)

1.1 Selection Sort

a) Sortieren Beispiel:

	3		15		22		2		37		7		111		9		12	
<hr/>																		
	2		15		22		3		37		7		111		9		12	
	2		3		22		15		37		7		111		9		12	
	2		3		7		15		37		22		111		9		12	
	2		3		7		9		37		22		111		15		12	
	2		3		7		9		12		22		111		15		37	
	2		3		7		9		12		15		111		22		37	
	2		3		7		9		12		15		22		111		37	
	2		3		7		9		12		15		22		37		111	

b) Vertausch-Operationen

$$\text{Vertausch Operationen} = \text{Anzahl der Elemente im Array} - 1$$

Anmerkung: Selbstvertauschung tritt auf, wenn das jetzige Element schon an der richtigen Stelle ist. Diese müssen per Hand aus der Tabelle von a) ausgelesen werden.

c) Vergleichs-Operationen

n: Anzahl der Elemente im Array

$$\text{Vergleichs Operationen} = \frac{n(n-1)}{2}$$

Anmerkung: Egal nach welchem Kriterium sortiert wird, diese Formel gilt immer!

1.2 Insertion Sort

a) Sortieren Beispiel

Original	6	4	10	12	2	15	19	5	14	9
Iteration 1	4	6	10	12	2	15	19	5	14	9
Iteration 2	4	6	10	12	2	15	19	5	14	9
Iteration 3	4	6	10	12	2	15	19	5	14	9
Iteration 4	2	4	6	10	12	15	19	5	14	9
Iteration 5	2	4	6	10	12	15	19	5	14	9
Iteration 6	2	4	6	10	12	15	19	5	14	9
Iteration 7	2	4	5	6	10	12	15	19	14	9
Iteration 8	2	4	5	6	10	12	14	15	19	9
Iteration 9	2	4	5	6	9	10	12	14	15	19

b) Vertausch-Operationen:

1. Markiere in jeder Zeile die „Vergleichs-Diagonale grün“
2. Markiere in jeder Zeile rot bis zum Index auf das grün markierte Element
3. Vertausch-Operationen = Anzahl der roten Elemente

c) Vergleichsoperationen:

1. Schritte 1 und 2 von „Vertausch-Operationen“
2. Markiere, wenn möglich, links ein zusätzliches Kästchen gelb
3. Vergleichs-Operationen = Anzahl der roten Elemente + Anzahl der gelben Elemente

1.3 Bubble Sort

a) Sortieren Beispiel

Original:	0	7	22	1	20	11	5	8	19	9	–
Iteration 1:	0	7	1	20	11	5	8	19	9	22	Swaps: 7
Iteration 2:	0	1	7	11	5	8	19	9	20	22	Swaps: 6
Iteration 3:	0	1	7	5	8	11	9	19	20	22	Swaps: 3
Iteration 4:	0	1	5	7	8	9	11	19	20	22	Swaps: 2
Iteration 5:	0	1	5	7	8	9	11	19	20	22	Swaps: 0

b) Vertausch-Operationen

Kein Muster, nur doch durchzählen möglich.

c) Vergleichsoperationen

*Vergleichs Operationen = selbstgeschrieben Zeilen * Elemente*

2 Sortieralgorithmen Divide and Conquer

2.1 QuickSort

Original:

[3, 8, 17, 5, 15, 2, 1, 12, 4, 9]

1. Pivotelement wählen und mit dem letzten Element tauschen.

[9, 8, 17, 5, 15, 2, 1, 12, 4, (3)]

2. Von **links** ausgehend Element suchen, das **größer** ist als das Pivotelement.

[**9**, 8, 17, 5, 15, 2, 1, 12, 4, (3)]

3. Von **rechts** ausgehend Element suchen, das **kleiner** ist als das Pivotelement.

[9 , 8, 17, 5, 15, 2, **1**, 12, 4, (3)]

4. Elemente aus 2. und 3. tauschen

[**1**, 8, 17, 5, 15, 2, **9**, 12, 4, (3)]

5. Schritt 2-4 solange wiederholen bis der Index des Elements aus 2. größer ist als der Index des Elements aus 3.

[**1**, **2**, 17, 5, 15, **8**, **9**, 12, 4, (3)]

6. Pivot Element an die richtige stelle zurücktauschen.

[1, 2, (3), 5, 15, 8 , 9, 12, 4, 17]

7. Liste am Pivot Element aufspalten

[1, 2] (3) [5, 15, 8 , 9, 12, 4, 17]