Westfälische Hochschule

Grundlagen der Informatik und Programmierung 1

Sortieren

Ordne das Chaos.

Prof. Dr. Tom Vierjahn

Fachbereich Wirtschaft und Informationstechnik
Westfälische Hochschule – Campus Bocholt

Wintersemester 2019/20

Lineare Suche

in unsortierten Daten

Daten:

finde die 13:

Aufwand:

Binäre Suche

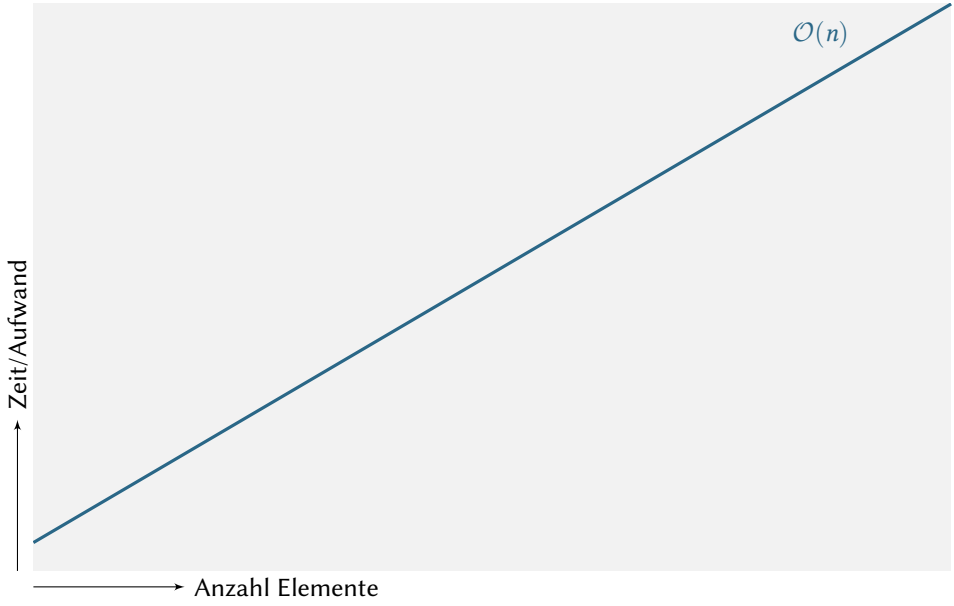
in sortierten Daten

Daten:

finde die 13:

Aufwand:

Laufzeit-Komplexität



Monkeysort

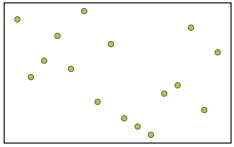
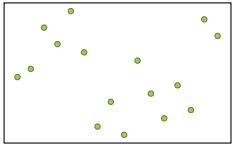
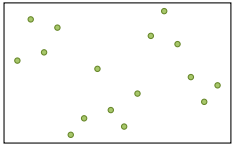
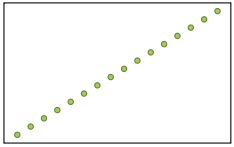
auch: Bogosort, Stupidsort, Permutationsort, Shotgunsort

Daten:

initial:	14	7	9	12	8	15	4	11	2	1	0	5	6	13	3	10
$i = 1$:	3	11	1	14	0	9	5	4	8	7	15	6	13	10	12	2
$i = 2$:	7	8	13	11	15	10	1	4	0	9	5	2	6	3	14	12
⋮																
$i = j$:	9	14	10	13	0	2	8	3	1	5	12	15	11	7	4	6
⋮																
$i = z$:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Monkeysort

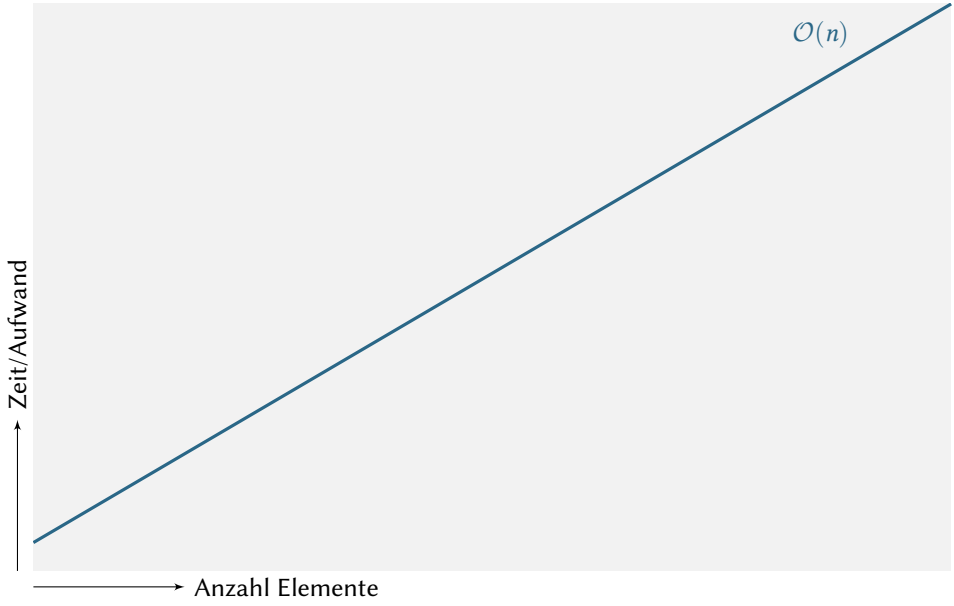
auch: Bogosort, Stupidsort, Permutationsort, Shotgunsort

Ablauf:

Aufwand:

Laufzeit-Komplexität



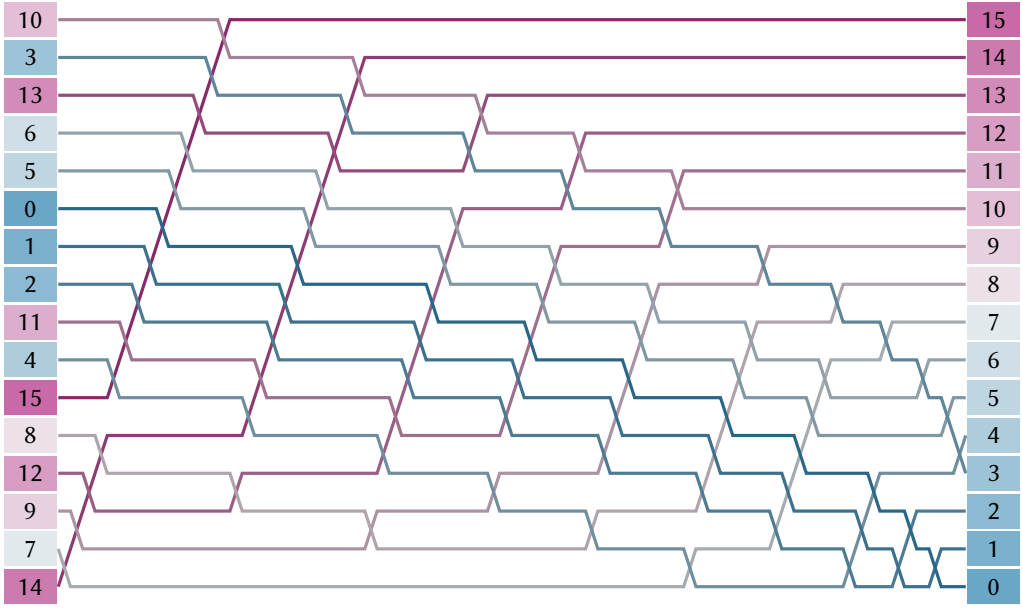
Bubblesort

Daten:

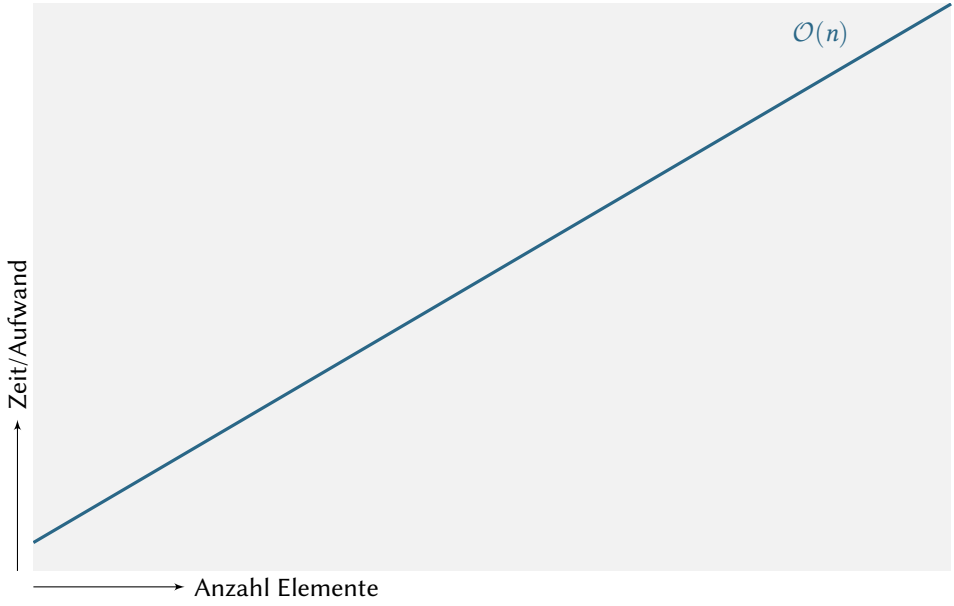
Ablauf:

Aufwand:

Bubblesort



Laufzeit-Komplexität



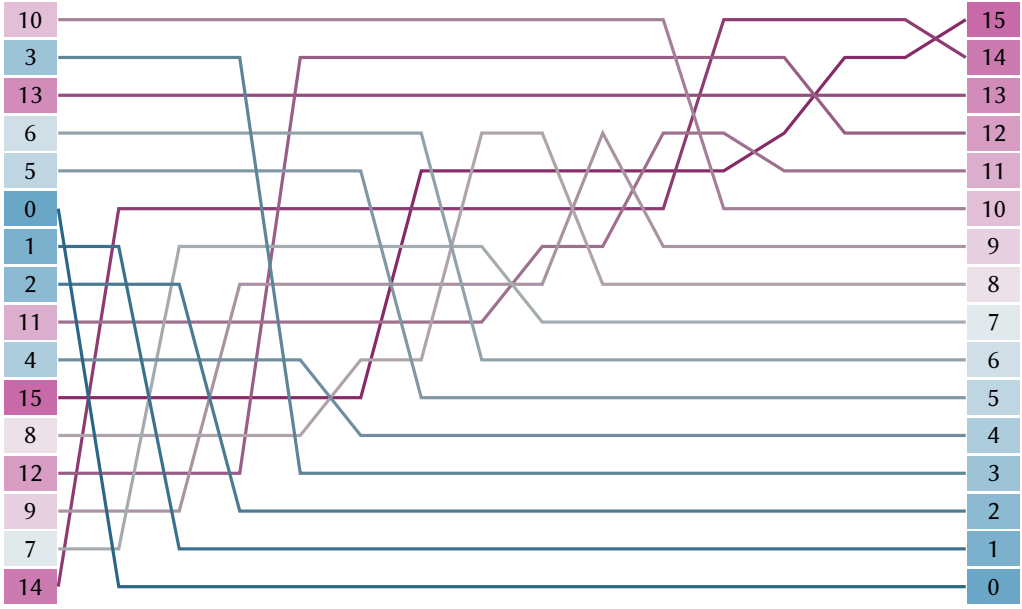
Selectionsort

Daten:

Ablauf:

Aufwand:

Selectionsort



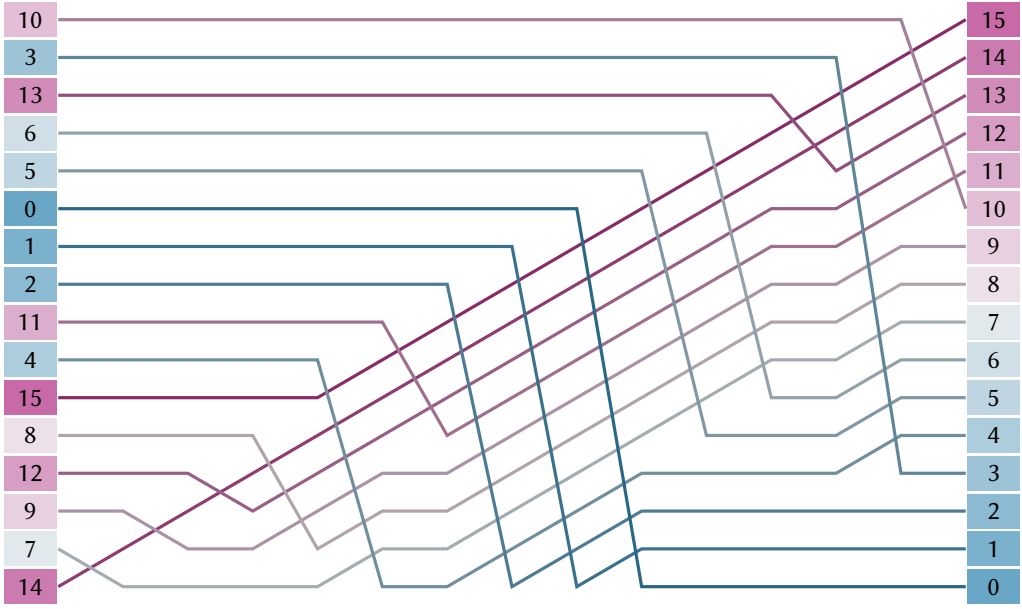
Insertionsort

Daten:

Ablauf:

Aufwand:

Insertionsort



Mergesort

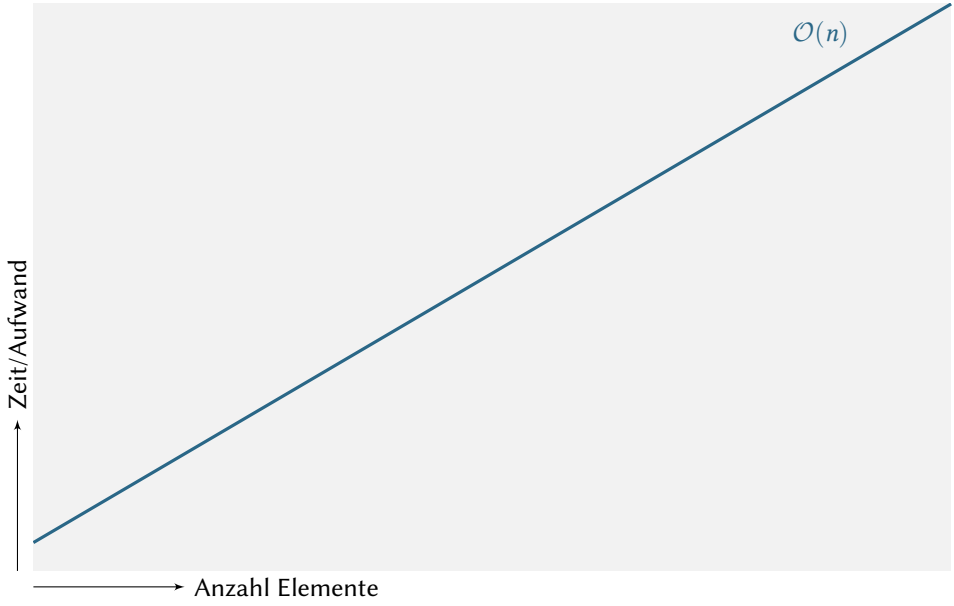
Daten:

Mergesort

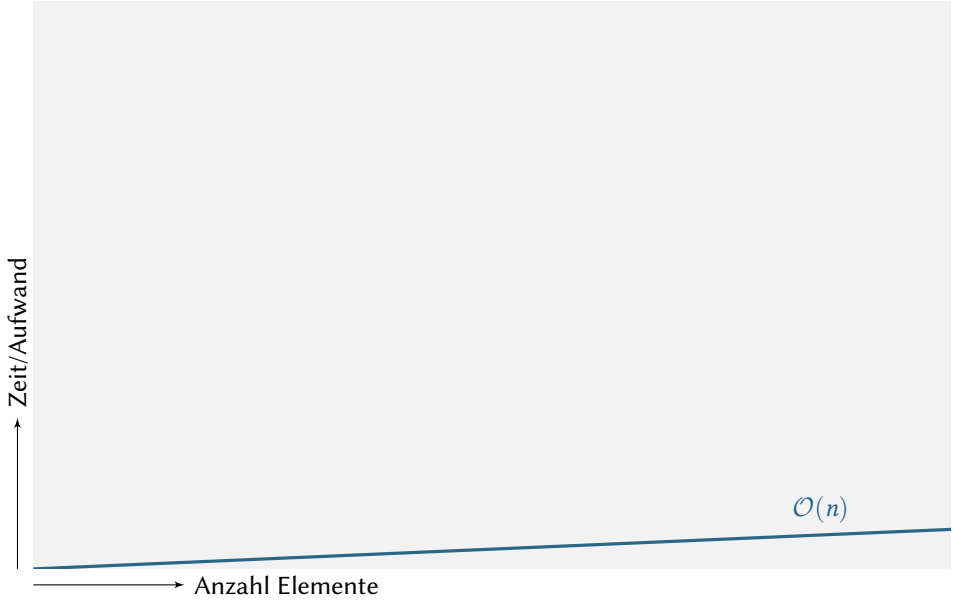
Ablauf:

Aufwand:

Laufzeit-Komplexität



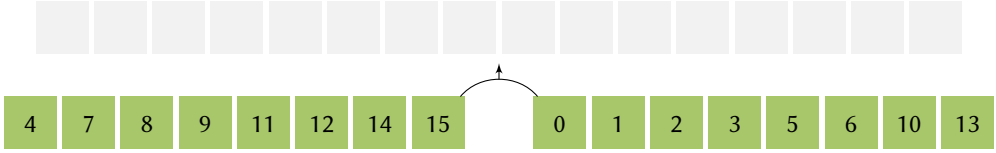
Laufzeit-Komplexität



Mergesort

sortiere beim Zusammenfügen

Daten:



Ablauf:

Aufwand:

Quicksort

Prinzip

Quicksort

Lomuto-Partition

Daten:

Ablauf:

Aufwand:

Quicksort

Hoare-Partition

Daten:

Ablauf:

Aufwand:

Minimaler Aufwand

Mögliche Sortierungen

- n Elemente können in Permutationen angeordnet werden.

Beispiel:

- 4 Elemente:

1	2	3	4	2	1	3	4	3	1	2	4	4	1	2	3
1	2	4	3	2	1	4	3	3	1	4	2	4	1	3	2
1	3	2	4	2	3	1	4	3	2	1	4	4	2	1	3
1	3	4	2	2	3	4	1	3	2	4	1	4	2	3	1
1	4	2	3	2	4	1	3	3	4	1	2	4	3	1	2
1	4	3	2	2	4	3	1	3	4	2	1	4	3	2	1

Minimaler Aufwand

Entscheidungen à la „Wer bin ich?“

$a_0 \,?\, a_1$				$\wedge$	$a_0 \,?\, a_2$				$\wedge$	$a_0 \,?\, a_3$				$\wedge$	$a_1 \,?\, a_2$				$\wedge$	$a_1 \,?\, a_3$				$\wedge$	$a_2 \,?\, a_3$			
1	2	3	4		2	1	3	4		3	1	2	4		4	1	2	3										
1	2	4	3		2	1	4	3		3	1	4	2		4	1	3	2										
1	3	2	4		2	3	1	4		3	2	1	4		4	2	1	3										
1	3	4	2		2	3	4	1		3	2	4	1		4	2	3	1										
1	4	2	3		2	4	1	3		3	4	1	2		4	3	1	2										
1	4	3	2		2	4	3	1		3	4	2	1		4	3	2	1										

Minimaler Aufwand

Herleitung

im Beispiel:

- 24 Permutationen: 6 Vergleiche

allgemeiner:

- starte mit $n!$ Permutationen
- Spieler nennt Vergleichspaar
- Gegner wählt Vergleichsoperator
  - möglichst viele Elemente übrig bleiben
  - maximal die Hälfte der Elemente wird aussortiert

Minimaler Aufwand

Herleitung

formaler:

- der Entscheidungsbaum hat Blätter
- ein Binärbaum der Höhe h hat Blätter
- für einen passenden Binärbaum:

Stabilität

Original:

	S
ALG	3
AT1	1
AT2	2
BSY	2
BWL	5
DBI	3
DCT	1

Zusammenfassung

	best.	mittel	schlecht.	stabil
Monkeysort	$\mathcal{O}(n)$	$\mathcal{O}(n \cdot n!)$	?	nein
Bubblesort	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$	ja
Selectionsort	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$	nein
Insertionsort	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$	ja
Mergesort	$\mathcal{O}(n \log n)$	$\mathcal{O}(n \log n)$	$\mathcal{O}(n \log n)$	ja
Quicksort	$\mathcal{O}(n \log n)$	$\mathcal{O}(n \log n)$	$\mathcal{O}(n^2)$	nein