

PROGRAMM: Sortieren 3.1

In der nächsten Version unseres Programms zum Sortieren von Zufallszahlen gibt uns das Programm die Zeit für alle Sortiermethoden aus. Die Methode `setTickCount()` gibt nach dem Sortieren die Zeit aus, die für die Methoden Insertion-Sort, Bubble-Sort und Quick-Sort gebraucht wurde.

z.B.:

Minimum: 1

Maximum: 10 000

Anzahl: 10 000

Insertion-Sort	20 s
Bubble-Sort	19 s
Quick-Sort	0,015 s

Danach haben wir uns die Zeiten im Verhältnis zur Menge der Daten angesehen. Unsere aufgestellte Vermutung für die Methode Insertion-Sort durch Ausprobieren mit verschiedenen Datenmengen war, dass $T \sim n^2$ ist. Dabei ist T die Laufzeit und n steht für die Menge der Daten. Um dies nachzuweisen haben wir uns den Algorithmus der Methode genauer angeschaut.

Algorithmus ist $O(n^2)$

	Daten n	Quellcode
1. Schleife:	1 - (n-1)	for i=0 to Anzahl -2
2. Schleife:	1 - n	for j=i+1 to Anzahl -1

Beispiel: n=10

Wert i	zu sortierende Zahlen		Durchläufe
i=0:	9	0-8	1-9
i=1:	8	0-8	2-9
i=2:	7	0-8	3-9
i=3:	6	0-8	4-9
:	:	:	:
i=8:	1	0-8	9-9

Anzahl der Schleifendurchläufe:

$$1 + 2 + 3 + \dots + 9 = \frac{9(9+1)}{2} = 45$$

Allgemein:

n	
i=0:	n-1
i=1:	n-2
:	:
i=n-1:	1

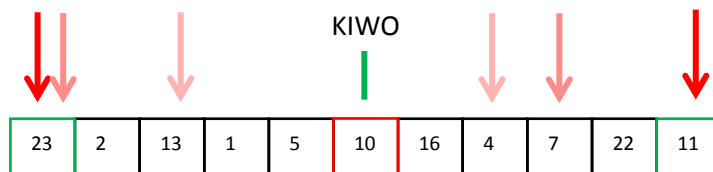
Anzahl der Schleifendurchläufe:
0

$$1+2+\dots+n+1 = 1+2+\dots+(n-1)+(n-n)$$

$$\begin{aligned} \int \frac{n(n+1)}{2} - n &= \frac{n^2 + n - 2n}{2} = \frac{n^2 * n}{2} = \frac{1}{2}n^2 - \frac{n}{2} \\ &\approx \frac{1}{2}n^2 * n \\ &\rightarrow \approx n^2 \end{aligned}$$

➔ Unsere Vermutung ist durch das Verallgemeinern und Vereinfachen des Beispiels bestätigt. Die Anzahl der Schleifendurchläufe steigt mit dem Quadrat der Menge der Daten.

Außerdem sind wir noch auf die neue Methode Quick-Sort eingegangen. Wie der Name schon sagt, sortiert diese Methoden die Daten deutlich schneller als die beiden anderen Methoden und kommt auch mit einer großen Datenmenge aus. Um eine Vorstellung von der Methode zu bekommen haben wir uns die Funktionsweise angesehen.



Zeiger fährt nach rechts bis eine Zahl **größer** als das Kiwo ist
 Zeiger fährt nach links bis eine Zahl **kleiner** als das Kiwo ist

BEIDE HABEN EINE ZAHL GEFUNDEN

➔ **TAUSCHEN**

➔ Vorsortierung

Alle kleiner ← 10 → Alle größer

