Komplexität

Um die Güte der behandelten Sortierverfahren beurteilen zu können, bestimmen wir jeweils die Anzahl der benötigten Vergleiche. Beim Sortieren durch Auswahl wird für i von 1 bis $n-1$ das Minimum im Array ab der Stelle i gesucht und an der i-ten Stelle eingefügt. Für $i = 1$ werden zum Finden des kleinsten Elements $n-1$ Vergleiche benötigt, zum Finden des zweitkleinsten $n-2$ Vergleiche und so fort, es sind also insgesamt:

$$(n-1) + (n-2) + \cdots + 2 + 1 \;=\; \frac{n(n-1)}{2} \;\approx\; \frac{n^2}{2} \qquad \text{Vergleiche erforderlich.}$$

Hierbei haben wir die Summenformel für die arithmetische Reihe benutzt: $\quad s_n \;=\; \dfrac{(a_1 + a_n)\cdot n}{2}$

Wir erhalten für das Sortieren durch Auswahl eine polynomiale Komplexität, genauer eine quadratische.

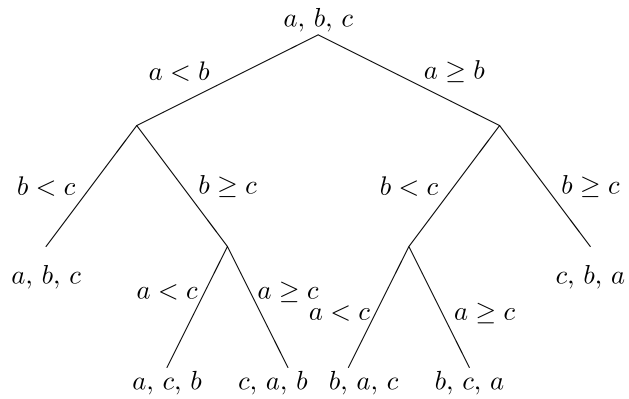
Für das Sortierverfahren Quicksort können wir die Anzahl der Vergleiche C_n für n Elemente nur näherungsweise bestimmen, eine exaktere Analyse brächte jedoch kein anderes Ergebnis.
Die Methode *partition* benötigt angenähert n Vergleiche. Da sie den Array in zwei Abschnitte unterteilt, die jeweils weiter rekursiv zu sortieren sind, ergibt sich für die Anzahl der Vergleiche C_n die rekursive Formel:

$$C_n \;=\; 2\cdot C_{\frac{n}{2}} + n$$

Falls z. B. $n = 32$ ist, erhalten wir $C_{32} \;=\; 2\cdot C_{16} + 32 \;=\; \cdots \;=\; 5\cdot 32$, wobei $C_1 = 0$ ist.

Da $2^5 = 32$ ist, also $5 = \log_2 32$, erhalten wir: Die Komplexität vom Quicksort ist von der Ordnung $n\cdot\log_2 n$.

Wir fragen uns, wie viele Vergleiche jedes Sortierverfahren mindestens benötigt. Betrachten wir hierzu das Sortieren von $n = 3$ Elementen. Den Sortiervorgang können wir uns durch einen Vergleichsbaum veranschaulichen. Jeder Sortiervorgang besteht aus einem Weg von der Wurzel zu einem der 6 Blätter, denn $3! = 1\cdot 2\cdot 3 = 6$.



Allgemein läßt sich auf diese Weise jedem Sortierverfahren ein binärer Entscheidungsbaum zuordnen, der mindestens $n!$ äußere Knoten (Blätter) hat. Denn n Elemente lassen sich auf $n!$ Weisen anordnen. Um die 1. Stelle zu besetzen, gibt es n Möglichkeiten, für die 2. Stelle $n-1$, usw., insgesamt gibt es $n(n-1)(n-2)\cdots 2\cdot 1 = n!$ Möglichkeiten, alle n Plätze zu besetzen.
Wie hoch ist dieser Baum mindestens, d. h., wie viele Vergleiche sind notwendig? Ein vollständiger Binärbaum der Höhe 3 hat 2^3 Blätter, ein Baum der Höhe k hat 2^k Blätter. Falls n Elemente sortiert werden, müssen die $n!$ möglichen Ergebnisse unter den 2^k Blättern sein, d. h. es muß gelten:

$$n! \;\le\; 2^k$$

Wie klein kann k gewählt werden? Antwort: $k = \log_2 n!$

Wir begnügen uns mit einer groben Abschätzung des logarithmischen Ausdrucks:

$$\log_2 n! \;=\; \log_2 1\cdot 2\cdot 3\cdot\cdots\cdot n \;\ge\; \frac{n}{2}\log_2\frac{n}{2} \approx n\cdot\log_2 n$$

Zusammenfassung:

Jeder Sortieralgorithmus benötigt näherungsweise mindestens $n\cdot\log_2 n$ Vergleiche.