

Wie schnell ist denn nun Sortieren?

Bisher haben wir für konkrete Sortierverfahren die Laufzeiten bestimmt und dabei erkannt, dass wir in quadratischer Zeit sortieren können. **Aber: Wie schnell kann ein Sortieralgorithmus im besten Fall sein?**

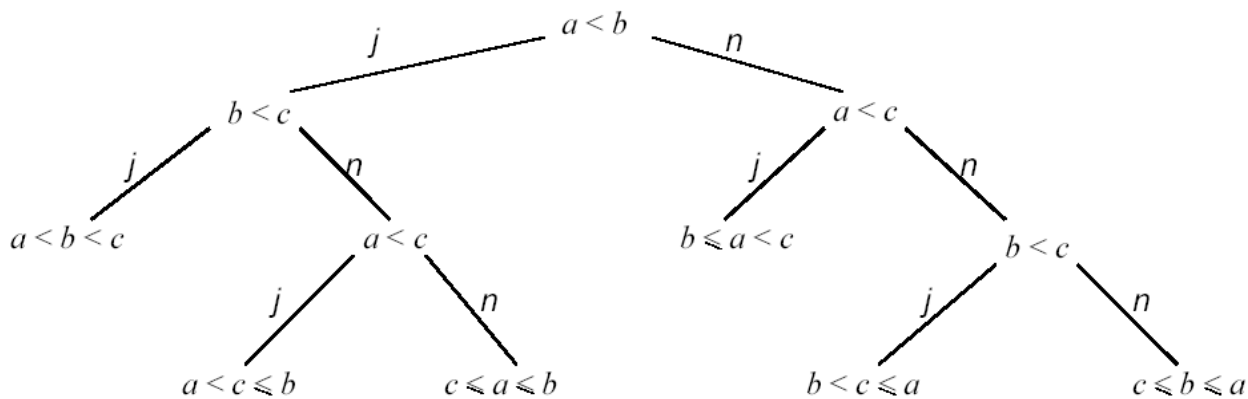
Dieser Frage muss man theoretisch begegnen, denn die Antwort muss ja allgemeingültig (also auch für Algorithmen, die vielleicht noch gar nicht entdeckt sind), beantwortet werden.

Beweis der unteren Schranke für vergleichsbasiertes Sortieren

Es lässt sich beweisen, dass ein vergleichsbasiertes Sortierverfahren nicht schneller als $n \cdot \log(n)$ sein kann:

Jedem möglichen Algorithmus, der mittels Vergleichen sortiert, kann ein so genannter Entscheidungsbaum zugeordnet werden. Dieser Entscheidungsbaum stellt in jedem Knoten eine der Vergleiche dar und jedes Blatt (also die unterste Ebene des Baumes) steht für eine Sortierreihenfolge der Elemente.

Beispiel eines Entscheidungsbaumes mit 3 Elementen a, b, c:



Ein Sortierverfahren muss prinzipiell in der Lage sein, aufgrund von Vergleichsentscheidungen alle diese möglichen Permutationen zu erzeugen.

Da alle Permutationen der zu sortierenden Zahlen das Ergebnis des Sortieralgorithmus sein könnten, muss der Entscheidungsbaum mindestens $n!$ Blätter haben.

In einem Entscheidungsbaum mit $n!$ Blättern beträgt die maximale und die mittlere Tiefe eines Blattes

mindestens $\log(n!)$. Da eine untere Schranke gesucht ist, kann $n!$ mittels $n! \geq \left(\frac{n}{2}\right)^{\frac{n}{2}}$ nach unten hin abge-

$$\log(n!) \geq \log\left(\left(\frac{n}{2}\right)^{\frac{n}{2}}\right) = \frac{n}{2} \cdot \log\left(\frac{n}{2}\right)$$

schätzt werden. Damit gilt