

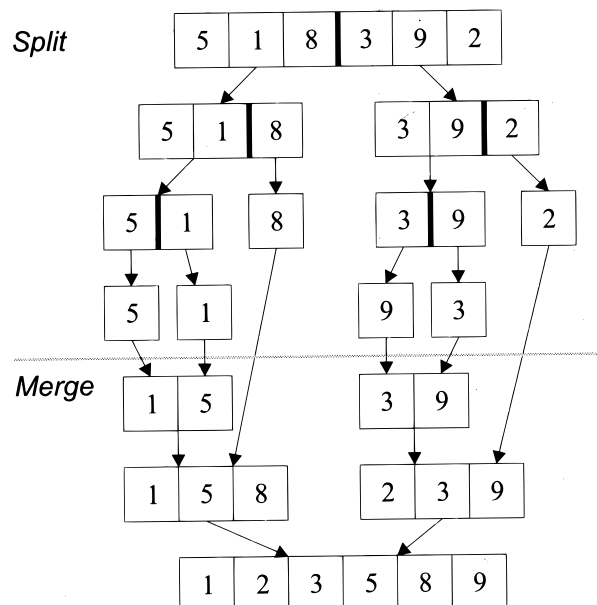
Allgemeines

In der Praxis kommt sehr häufig das Problem vor, dass große sortierte Datenbestände in zwei Dateien zusammengefasst und sortiert in eine zusammenfassende Datei geschrieben werden müssen. Beispiel: Zwei Banken fusionieren und die sortierten Kundendaten beider Institute werden in einer neuen Datei zusammen gefasst. Eine einzelne Datei kann dabei so groß sein, dass sie nicht in den Hauptspeicher eines Computers passt. Man liest dann die beiden Dateien datensatzweise vom Anfang bis zum Ende und schreibt jeweils das kleinste Element in die neue Datei. Diesen Vorgang bezeichnet man als **Mischen**. Der Begriff hat nichts mit dem *Mischen* bei einem Kartenspiel, was ja einer Rückgängigmachung der Sortierung entspricht, zu tun.

Das Mischen kann auch bei der internen Sortierung von Feldern benutzt werden.

Idee zum Algorithmus

Man teilt das Feld in zwei Teile, sortiert diese einzeln und mischt die beiden Teile wieder zum Gesamtfeld zusammen. Nach dem gleichen Prinzip sortiert man die Teilfelder rekursiv, bis auf der Ebene von Feldern mit nur einem Element nur noch das Mischen übrigbleibt. Das Prinzip wird in der folgenden Graphik dargestellt:



Programmierung

Für das Mischen wird ein gleichgroßes Hilfsfeld benötigt. Die sortierten Teile werden in das Hilfsfeld geschrieben und daraus zurück in das eigentliche Feld gemischt. Dabei zeigt es sich, dass das Mischen deutlich einfacher zu formulieren ist, wenn man den zweiten Teil umgekehrt sortiert in das Hilfsfeld schreibt.

Grobformulierung des Algorithmus’:

Sortiere von `links` bis `rechts`

- Sortiere von `links` bis `mitte`.
- Sortiere von `mitte+1` bis `rechts`.
- Schreibe linken Teil in das Hilfsfeld.
- Schreibe rechten Teil umgekehrt in das Hilfsfeld.
- Schreibe durch Mischung die beiden Teile aus dem Hilfsfeld zurück ins Feld.

Formuliere diesen rekursiven Algorithmus als Java-Methode.

Testphase und Präsentation

Teste den Algorithmus nicht nur an Feldern mit zufälligen Inhalten, sondern auch an „fast schon“ sortierten Feldern und umgekehrt sortierten Feldern. Stelle das Zeitverhalten des Algorithmus graphisch dar und stelle fest, wie die benötigte Zeit von der Anzahl der Elemente n abhängt. Vergleiche die verschiedenen Fälle mit den anderen Sortieralgorithmen.

Die Messergebnisse aller bisher behandelten Sortieralgorithmen sollen in einer HTML-Datei geeignet dargestellt bzw. präsentiert werden.