



Analyse einer Sortierverfahrens: Mergesort

Name:

Vorname:

Klasse:

Mergesort – Sortieren durch Mischen

Das Verfahren nutzt das informatische Prinzip „**Teile und Herrsche**“, bei dem ein Problem in kleinere Probleme zerlegt wird. Diese können dann einzeln gelöst werden. Gegebenenfalls muss das Teilproblem weiter zerlegt werden, um eine Teillösung zu erhalten.

Der **prinzipielle Ablauf** beim Sortieren durch Mischen erfolgt in 3 Schritten:

1. Zerlegung der Liste in zwei Teillisten
2. Sortierung der beiden Teillisten
3. Mischen der Teillisten zu einer Liste:

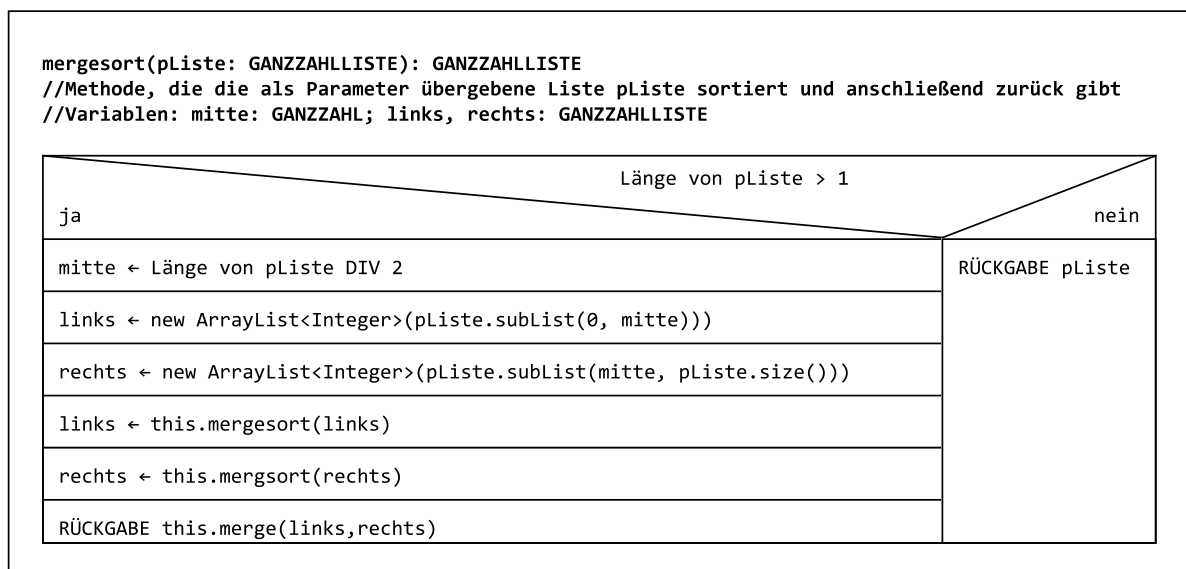
Es wird das erste Element der ersten Teilliste mit dem ersten Element der zweiten Teilliste verglichen und das kleinere der beiden in die Gesamtliste übernommen. Dieser Vorgang wiederholt sich, bis keine Elemente mehr in den Teillisten vorhanden sind. Sollte eine Teilliste leer sein, können alle Elemente der anderen Teilliste zur Gesamtliste hinzugefügt werden.

Dieser Ablauf wird **rekursiv** ausgeführt, d. h. die Zerlegung (Schritt 1) wird so oft fortgesetzt, bis die Liste aus nur einem Element besteht. Dann ist sie bereits sortiert.

Aufgaben:

- 1) Legen Sie mit den Spielkarten eine Reihe mit sechs Elementen in worst-case-Sortierung. Sortieren Sie unter Anleitung und zählen Sie dabei die Anzahl der Vergleiche.
- 2) Vergleichen Sie die Werte der Vergleiche mit den Daten vom Arbeitsblatt Ripplesort/Bubblesort und leiten Sie eine Vermutung ab.

Darstellung des Algorithmus als Struktogramm





Analyse einer Sortierverfahrens: Mergesort

Name:

Vorname:

Klasse:

```
merge(pLinks, pRechts: GANZZAHLLISTE): GANZZAHLLISTE
//Mischmethode mit Rückgabe der gemischten Teillisten
//Variablen: i, j: GANZZAHL; gemischt: GANZZAHLLISTE
```

gemischt ← neue GANZZAHLLISTE	
i ← 0	
j ← 0	
SOLANGE (i < Länge von pLinks UND j < Länge von pRechts)	
i-tes Element von pLinks <= j-tes Element von pRechts	
ja	nein
füge i-tes Element von pLinks zu gemischt hinzu	füge j-tes Element von pRechts zu gemischt hinzu
inkrementiere i	inkrementiere j
erhöhe Vergleichszähler	
SOLANGE i < Länge von pLinks	
füge i-tes Element von pLinks zu gemischt hinzu	
inkrementiere i	
SOLANGE j < Länge von pRechts	
füge j-tes Element von pRechts zu gemischt hinzu	
inkrementiere j	
RÜCKGABE gemischt	

Aufgabe:

Implementieren Sie den Algorithmus in die gegebene Java-Vorlage.

Experimente zur Untersuchung des Algorithmus

Prüfen Sie Ihre Vermutung aus Aufgabe 2. Analysieren Sie das Laufzeitverhalten des Algorithmus in Analogie zum Arbeitsblatt Ripplesort. Wählen Sie die Elementzahl so, dass die Sortierzeit mindestens 100 ms beträgt.