

Aufgabe 3: Sortieren II

In dieser Aufgabe werden zwei unterschiedliche Typen von Algorithmen verglichen: rekursive und randomisierte Algorithmen.

Aufgabenstellung

Sei `zahlen` eine endliche Folge von ganzen positiven Zahlen (in Form einer Liste/Reihung) gegeben. Die ist nach der Grösse zu sortieren (Kleinstes Element vorne bzw. links). Verwenden Sie dazu die im ersten Praktikum implementierte ADT `array` (Achtung: ggf. wird die Datei `array.erl` mit Implementierungen anderer Gruppen ausgetauscht!).

Implementieren Sie nun folgende **zwei Algorithmen**:

quicksortRekursiv als **rekursive Lösung**:

Dieser Algorithmus ist rekursiv und mit dem Pivotelement an linkerster Stelle zu implementieren. Sind weniger als **12 Elemente** (innerhalb der Rekursion) zu sortieren, ist eines der Programme aus Praktikumsaufgabe zwei zu verwenden, d.h. ab einer Anzahl von 12 Elementen arbeitet das Programm rekursiv und verzweigt bei weniger als 12 Elementen (für genau diese Elemente!) auf das andere Programm.

quicksortRandom als **randomisierte Lösung**: Dieser Algorithmus ist rekursiv und mit einer Zufallsauswahl des Pivotelementes zu implementieren. Sind weniger als **12 Elemente** zu sortieren, ist genau so wie bei der rekursiven Lösung zu verfahren.

Die Zahlen sind aus einer Datei `zahlen.dat` einzulesen. Die sortierten Zahlen sind in eine Datei `sortiert.dat` auszulesen. Versuchen Sie möglichst viele Zahlen zu sortieren.

Alle Algorithmen sind mit Zeitmessungen zu versehen.

Die Anzahl der Vergleiche und Verschiebungen der Elemente ist zu zählen. Dies sollte nach der Zeitmessung in eine Kopie der implementierten Algorithmen eingebaut werden, um bei der Zeitmessung den Aufwand fürs Zählen nicht mit zu berücksichtigen, auf der anderen Seite aber nicht zu früh zwei Kopien pflegen zu müssen!

Die Algorithmen sind 100-mal auszuführen, wenn möglich mit jeweils unterschiedlichen Zahlen: 80-mal Zufallszahlen und jeweils 10-mal "best case" bzw. "worst case" Zahlen. Aus den Zeitmessungen ist dann per Mittelwert anzugeben:

Anzahl eingelesener Elemente

Name des Algorithmus

Benötigte Zeit, sowie maximal und minimal benötigte Zeit.

Anzahl Vergleiche, sowie maximale Anzahl und minimale Anzahl an Vergleichen.

Anzahl Verschiebungen, sowie maximale Anzahl und minimale Anzahl an Verschiebungen.

Die Daten sind in einer Datei messung.log zu speichern. Zusätzlich, zur statistischen Aufbereitung, können auch *.csv dateien erzeugt werden.