Programmiertechnik II
Angewandte Informatik
SS 2018
Prof. Dr. Oliver Bittel

HOCHSCHULE KONSTANZ
TECHNIK, WIRTSCHAFT UND GESTALTUNG

Aufgabenblatt 8 – Sortierverfahren

Generische Sortiermethoden

Implementieren Sie die beiden folgenden Sortierverfahren als generische Methoden mit Typbeschränkung und testen Sie die Verfahren für Integer- und String-Felder.

1. **Hybrides QuickSort:**

   Das Sortierverfahren arbeitet wie QuickSort. Sobald die Größe des Teilfelds jedoch kleiner als z.B. N = 100 wird, wird mit insertionSort sortiert. Es ist hilfreich, insertionSort so anzupassen, dass ein Teilfeld a[li], ..., a[re] sortiert werden kann:

   ```
   void insertionSort(T[] a, int li, int re)
   ```

2. **Hybrides QuickSort mit 3-Median-Strategie:**

   Erweitern Sie das Verfahren aus 1. durch die 3-Median-Strategie.

Laufzeitmessungen

Führen Sie Laufzeitmessungen für die beiden implementierten Verfahren durch und vergleichen Sie diese mit der der generischen Sortiermethode aus java.util.Arrays. Ermitteln Sie Laufzeiten für Felder mit n = 100_000 und n = 200_000 Elementen für folgende Fälle:

- Felder mit zufälligen ganzen Zahlen. Mit

  ```
  (int) (Math.random() * M)
  ```

  lassen sich gleichverteilte zufällige ganze Zahlen aus [0,M) erzeugen.
- Felder mit sortierten Zahlen.

Binäre Suche

Schreiben Sie auf Basis des Algorithmus der binären Suche eine Methode rangeSearch, die in einem sortierten Feld a die Anzahl der Elemente ermittelt, die sich in einem Intervall [u, v) befinden.

```
int rangeSearch(T[] a, T u, T v)
```

Testen Sie Ihre Methode mit dem Testprogram auf der Web-Seite.

Initialisieren Sie ein Feld a der Größe n = 100_000 mit zufälligen ganzen Zahlen kleiner als M = 1000 und sortieren Sie das Feld a. Prüfen Sie dann, ob die Anzahl der Zahlen aus [250, 750) im Feld a erwartungsgemäß etwa n/2 beträgt.