

Übungsblatt 1

Ausgabe: 21.3.2020

Abgabe: 1.4.2020 (7:00 Uhr)

Testat: 3.4.2020 (9:45 - 13:40 Uhr)

Aufgabe 1: Suchfunktionen - *Programmieraufgabe* -

30 Punkte

Schreiben Sie vier **static** Suchfunktionen, die einen Wert in einem Array suchen und den Index des Wertes (falls vorhanden) oder -1 (falls nicht vorhanden) zurückliefern:

- Lineare Suche**, wobei immer das gesamte Array durchsucht wird und der Index des letzten gefundenen Wertes zurückgegeben wird.
- Lineare Suche**, wobei die Funktion zurückgekehrt, sobald der Wert gefunden wurde.
- Binäre Suche** in der **rekursiven** Fassung.
- Binäre Suche** in der **iterativen** Fassung.

Ihre Funktionen werden mit den bereitgestellten JUnit-Tests getestet. Sie müssen als kein Hauptprogramm schreiben. Wenn Sie wollen dürfen Sie das natürlich auch machen.

Denken Sie an Kommentare!

Anmerkung: Die lineare Suche gemäß a) wird z.B. benötigt, wenn Sie bei mehrfach vorkommenden Werten den Index des letzten haben möchten. Die lineare Suche nach b) liefert demgegenüber den ersten Index des gesuchten Wertes.

Aufgabe 2: Komplexitätsexperiment - *Programmieraufgabe* -

30 Punkte

Ändern Sie die Suchfunktionen aus Aufgabe 1 so um, dass Sie die **Anzahl der Schlüssel-Vergleiche** beim Suchen zählen können. Die Funktionen sollen zu diesem Zweck nicht den Index des gefundenen Elements, sondern die **Anzahl der Vergleiche** zurückliefern.

Ermitteln Sie die Anzahl der Schlüssel-Vergleiche

- bei erfolgreicher und erfolgloser Suche
- für alle vier Suchfunktionen
- für drei verschiedene Array-Größen (Länge jeweils verdoppelt, z.B. 1024 – 2048 – 4096 Elemente)

Tragen Sie die Ergebnisse in einer **Tabelle** ein, z.B.

		Größe des Arrays		
		1024	2048	4096
erfolgreich	Suchverfahren			
	linear (erster Treffer)			
	linear (letzter Treffer)			
	binär (iterativ)			
erfolglos	binär (rekursiv)			
	linear (erster Treffer)			
	linear (letzter Treffer)			
	binär (iterativ)			
	binär (rekursiv)			

und diskutieren Sie die Ergebnisse: welchen Einfluss hat die Array-Größe oder die Strategie? Können Sie die Zahlen erklären, gibt es ggf. Abweichungen gegenüber von Ihnen erwarteten Werten? Um eine **statistisch signifikante Aussage** für erfolgreiche Suche zu erhalten, müssen Sie die Suche mehrmals durchführen und anschließend den Mittelwert bestimmen. Damit die Suche erfolgreich ist, finden Sie unten Hinweise zur Belegung des Arrays und zur Wahl der Schlüssel.

- Suchen Sie jeweils mindestens 100 mal
- mit zufällig gewählten Schlüsseln

Wie sieht es bei der erfolglosen Suche, aus? Sie können die gesuchten Werte wie bei der erfolgreichen Suche bestimmen (wie müssten Sie das Experiment ändern?); aber ist das Experiment überhaupt nötig, oder können Sie die gesuchten Werte auch errechnen?

Hinweise:

- Kopieren Sie Ihre Programme aus Aufgabe 1, bevor Sie sie für diese Aufgabe verändern!
- Dokumentieren Sie Hauptprogramm und die Suchfunktionen mit sinnvollen Kommentaren.
- Für das Experiment „erfolgreiche Suche“ müssen Sie
 - 1) ein Array mit sortierten Zahlen definieren, und
 - 2) alle Zahlen aus dem Array mit der gleichen Wahrscheinlichkeit suchen, sowie
 - 3) immer erfolgreich suchen.

Dies können Sie einfach erreichen, indem Sie

- für das sortiertes Array die Array-Elemente mit Index als Wert belegen, d.h.

```
for (int i = 0; i < data.length; i++) data[i] = i;
```

(ein automatisches Belegen ist bei einem 1024 Elemente großen Array natürlich sinnvoll),

- oder als Schlüssel zufällig eine Zahl aus dem Bereich 0 bis N-1 (einschließlich) wählen; zufällige ganzzahlige Werte im Bereich 0 bis N-1 können Sie in Java erzeugen mit der Funktion:

```
(int) Math.floor(Math.random()*N)
```