

## Trainingsaufgaben Heap & HeapSort

### Motivation

Diese Aufgaben ergänzen die Übungsaufgaben, um Ihnen die Möglichkeit zu zusätzlichem Training zu geben. Die Lösungen finden Sie in einem separaten Dokument.

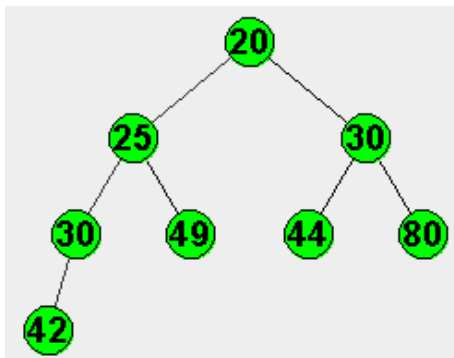
#### 1. Heap-Aufbau nach Floyd

Das abgebildete Array soll durch Anwendung des „*Algorithmus von Floyd*“ zu einem gültigen Max-Heap aufgebaut werden. Verwenden Sie das abgebildete Array direkt als Heap-Array und führen Sie darauf den Algorithmus auf. Sie können dies entweder direkt im Array erledigen oder den Heap zur Hilfe als Baumstruktur aufzeichnen. Zeigen Sie nach erfolgreichem Max-Heap Aufbau nochmals, wie die Elemente nun im Heap-Array angeordnet sind.

5	21	54	4	21	10	97	85	48	7	2	66	7	32	91	46	25
---	----	----	---	----	----	----	----	----	---	---	----	---	----	----	----	----

#### 2. Operationen auf einem bestehenden Heap

Führen Sie auf dem abgebildeten Min-Heap die unten angegebenen Operationen in entsprechender Reihenfolge aus:



removeMin(), add(1), add(5), add(3), add(49), add(50), add(-1), removeMin(), removeMin()

**Hinweis:** Mit Hilfe der Übung *HeapTest* können Sie selbständig Operationen auf MinHeaps üben. Sollte Ihre selbständig programmierte Variante noch nicht einwandfrei funktionieren, können Sie die Lösungen auf dem Netzlaufwerk verwenden.

#### 3. HeapSort: Baum Zeichnung

Gegeben sei ein Array, welches in einen „Heap-Bereich“ (Max-Heap) und einen „sortierten Bereich“ unterteilt ist. Es wurden bereits einige Sortier-Iterationen ausgeführt. Geben Sie an, wie viele Elemente der „sortierte Bereich“ umfasst.

Array:

25	18	16	4	17	15	7	1	25	92	135
----	----	----	---	----	----	---	---	----	----	-----

Zeichnen Sie den Heap als Baumstruktur. Sortieren Sie nun die restlichen Elemente Schritt für Schritt mit dem HeapSort Algorithmus.

#### **4. HeapSort: Array**

Gegeben ist ein vollständig aufgebauter Min-Heap. Führen Sie den HeapSort Algorithmus direkt auf dem Array aus, so dass die Elemente schlussendlich in absteigender Reihenfolge sortiert sind. Beschriften Sie dazu zuerst die Array-Indizes. Wählen Sie als Index für das erste Element 0 oder 1, je nachdem, welche Heap-Implementation Sie bevorzugen.

12	20	35	42	50	37	39
----	----	----	----	----	----	----

#### **5. HeapSort: Array**

Betrachten Sie Ihre (oder unsere) Lösung der Selbststudiums-Aufgabe zu HeapSort mit dem SortDemo-Framework. Wo müssen Sie überall etwas ändern – und was, damit das Programm absteigend und nicht aufsteigend sortiert?