

Übung 05: Heapsort, Countingsort

Aufgabe 1: Datenstruktur Heap

Zeichnen Sie alle möglichen (gültigen) MaxHeaps, die die 5 Schlüssel AAABB enthalten. Gehen Sie davon aus, dass der Buchstabe „B“ größer ist als der Buchstabe „A“.

Aufgabe 2: Heapsort¹

Gegeben sei ein Array A bestehend aus den folgenden Zeichen: $A = \langle X, C, B, D, M, S, Z, U, F \rangle$.

Sortieren Sie in dieser Aufgabe schrittweise mit Heapsort **alphabetisch aufsteigend**, so dass das Ergebnis lautet: $\langle B, C, D, F, M, S, U, X, Z \rangle$.

Hinweis: Verwenden Sie wie in der Vorlesung einen Max-Heap bei dem der „größte“ Buchstabe an der Wurzel steht. Beispielsweise ist B größer als A .

Gehen Sie in der folgenden Reihenfolge vor:

- Illustrieren Sie die Funktionsweise der Operation $\text{BUILD-MAX-HEAP}(A, 9)$ anhand des vorgegebenen Arrays A !
Es genügt, wenn Sie die Belegung des Arrays A in „Arraydarstellung“ **nach jeder einzelnen Vertauschung** angeben. Es wird aber dringend empfohlen, zur Arraybelegung die dazugehörige Baumstruktur zu zeichnen.
- Illustrieren Sie den eigentlichen Heapsort-Algorithmus², nachdem wie in Aufgabe a) ein initialer *MaxHeap* erzeugt wurde.
Geben Sie dazu die Belegung des Arrays A **nach jeder einzelnen Vertauschung** an!
Empfehlung: Dringend Baumstruktur mitzeichnen!

Aufgabe 3: Countingsort³

Countingsort ist ein Sortierverfahren, das nicht auf Vergleichen beruht. Dabei muss der Wertebereich der zu sortierenden Schlüssel klein und bekannt sein. Implementieren Sie die folgende Funktion:

```
public static Integer[] sort(Integer[] input, Integer k) {
```

Ein übergebenes Integer-Array `input` wird mit einem **stabilen** Countingsort-Verfahren sortiert, siehe Vorlesungsunterlagen. Bekannt ist, dass alle Zahlen in `input` den Wertebereich **[0..k]** haben. Das Input-Array darf nicht verändert werden. Für das Ergebnis wird ein neues Array erzeugt und zurückgegeben.

Sie können das mitgelieferte Codegerüst⁴ verwenden. Die Signatur der Methode darf nicht verändert werden. Ebenso wird eine Testklasse mitgeliefert.

¹ Animation: <https://www.cs.usfca.edu/~galles/visualization/HeapSort.html>

² Zeile 2-4 des Pseudocodes von $\text{HEAPSORT}(A, n)$ auf Folie 40.

³ Animation: <https://www.cs.usfca.edu/~galles/visualization/CountingSort.html>

⁴ Gitlab: Uebungen/Uebung05. Dort sind sowohl die java-Dateien als auch ein komplettes IntelliJ Projekt vorgegeben.