

7. Übungsblatt zu Algorithmen I im SoSe 2016

<https://crypto.iti.kit.edu/index.php?id=algo-bose16>
 {lisa.kohl,lukas.barth}@kit.edu

Aufgabe 1 (Binäre Heaps, 2 Punkte)

Nehmen Sie die letzten 4 Ziffern Ihrer Matrikelnummer und bilden Sie daraus alle möglichen binären Min-Heaps. Stellen Sie dabei die Heaps als implizites Feld dar.

Aufgabe 2 (Radixsort, 3 Punkte)

Führen Sie Least-Significant-Digit (LSD) Radixsort mit Dezimalziffern (Radix $K = 10$) auf dem folgenden Array durch. Verwenden Sie dazu als Untermethode die Methode *KSortArray*. Tragen Sie nach jeder Runde das Array in eine der Schablonen ein, markieren Sie die *Bucketgrenzen der Ziffern* durch Trennstriche im Array. Geben Sie zusätzlich jeweils den Zustand des Hilfsarrays c nach Durchlauf der zweiten for-Schleife in der Methode *KSortArray* an (vergleiche Vorlesung 30.05, Folie 5, $c := [1, 3, 5, 6]$).

Eingabe:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
31	13	64	52	91	69	19	7	57	29	58	78	9	91	11	44

Hilfsarray c :

0	1	2	3	4	5	6	7	8	9

Ergebnis nach der 1. Runde:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Hilfsarray c :

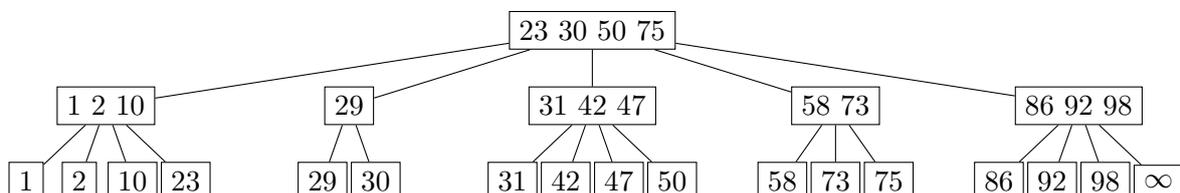
0	1	2	3	4	5	6	7	8	9

Ergebnis nach der 2. Runde:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Aufgabe 3 ((a, b)-Bäume, 6 Punkte)

Führen Sie auf dem folgenden $(2, 5)$ -Baum die geforderten Operationen in der angegebenen Reihenfolge durch. Zeichnen Sie den Endzustand des Baums nach jeder der angegebenen Operationen.



- a) Einfügen von 43, Einfügen von 45, Einfügen von 38.
- b) Beginnen Sie wieder *beim Ausgangszustand*. Löschen von 50, Löschen von 29, Löschen von 47.

Aufgabe 4 (*d*-äre Heaps, 1 + 2 + 1 + 3 Punkte)

In Vorlesung und Übung sind binäre Min-Heaps vorgestellt worden, also Heaps, bei dem jeder innere Knoten zwei größere (oder gleichgroße) Kinder hat. Analog dazu definieren wir einen *d*-ären Min-Heap als einen Min-Heap, bei dem jeder innere Knoten *d* größere (oder gleichgroße) Kinder hat.

- a) Sei zunächst $d = 3$. Erstellen Sie einen 3-ären Min-Heap aus folgenden Zahlen:

7, 59, 13, 47, 3, 17, 73, 19, 91, 71, 37, 83, 41

- b) Sei nun wieder *d* unbestimmt. Wie würden Sie allgemein einen *d*-ären Min-Heap als Array darstellen? Geben Sie Funktionen zum Finden der Kinder und des Elternknotens eines Elements an.
- c) Wie ist die Höhe eines *d*-ären Heaps mit *n* Elementen?
- d) Geben Sie eine effiziente Implementation von *deleteMin* für einen *d*-ären Heap in Pseudocode an. Sie dürfen hierzu die Funktionen aus Teilaufgabe b) benutzen. Geben Sie die Laufzeit von *deleteMin* bezüglich *n* und *d* an.

Ausgabe: Mittwoch, 08.06.2016

Abgabe: Freitag, 17.06.2016, 12:45 im Briefkasten im Untergeschoss von Gebäude 50.34

Deckblatt Übungsblatt 7

Algorithmen I

A1	A2	A3	A4	Σ
-----------	-----------	-----------	-----------	----------

Tutoriumsnummer:

Name

Matrikelnummer

Unterschrift

_____	_____	_____
_____	_____	_____

Mit unseren Unterschriften bestätigen wir, dass die Aufgaben von den Unterzeichnern eigenständig gelöst worden sind.

Hinweis: Das Übungsblatt darf in Gruppen von bis zu zwei Personen bearbeitet werden. Beide Personen müssen demselben Tutorium zugeteilt sein. Möchte jemand seine Abgaben-Gruppe innerhalb des Semesters wechseln, so ist dies im Voraus mit dem Tutor abzusprechen. **Bitte tragen Sie in das obere Quadrat groß die Nummer Ihres Tutoriums ein.** Die Lösung des Übungsblattes ist in jedem Fall mit diesem Deckblatt abzugeben.