

Aufgabe 5.1

Sortiere die Folge $\langle 365, 102, 864, 928, 380, 361, 635 \rangle$ mit:

- LSD-Radixsort mit $K = 10$. Gib für jede Iteration die Behälter von $kSort$ und die Reihenfolge der Einträge nach der Iteration an.
- MSD-Radixsort mit $K = 10$. Gib die Verteilung auf die Behälter an.

Aufgabe 5.2 (4 Punkte)

Gegeben sei ein leerer Heap.

- Füge die Elemente 15, 23, 12, 6, 13, 9, 2 und 10 in dieser Reihenfolge ein und stelle diesen nach jedem Schritt grafisch dar. Stelle den entstandenen Heap am Ende auch in einem Array dar.
- Entferne aus dem resultierenden Heap zweimal das Minimum und stelle den sich ergebenden Heap jeweils dar.

Aufgabe 5.3 (4 Punkte)

Beschreibe einen Algorithmus, der k sortierte Listen mit Gesamtlänge n in $O(n \log k)$ Zeit zusammenfügt zu einer sortierten Liste.

Hinweis: Benutze einen Heap der Größe k .

Aufgabe 5.4 (4 Punkte)

Gegeben sei die folgende alternative Prozedur zum Erstellen eines binären Heaps für ein unsortiertes Array $A[1..n]$:

buildHeapInsert($A : \text{Array}$)

```
1 for  $i := 1$  to  $n$ 
2   do insert( $A[i]$ )
```

- Zeige, dass *buildHeapInsert* im schlechtesten Fall die Laufzeit $\Theta(n \log n)$ benötigt, um einen Heap mit n Elementen zu erzeugen.
- Erzeugen die Prozedur *buildHeapBackwards* aus der Vorlesung und *buildHeapInsert* immer den gleichen Heap, wenn sie auf das gleiche Array angewendet werden? Beweise, dass sie dies tun, oder gib ein Gegenbeispiel an.