

# Aufgaben zur Veranstaltung

## Algorithmen und Datenstrukturen, SS 2022

H. Pflug, J. Dietel

FH Aachen, Campus Jülich; IT Center, RWTH Aachen

### Klausurvorbereitung, Blatt 2

27.06.2022

#### Probeklausur

##### Aufgabe 1

Notieren Sie die Antworten in ein oder mehreren Stichworten oder einem kurzen Satz:

- a) Die Java-Klasse für eine doppelt verkettete Liste heißt:
- b) Die Java-Klasse für eine Hashtabelle heißt:
- c) Insertion-Sort hat im Best-Case die minimale Laufzeitkomplexität:
- d) Die drei Grundoperationen für reguläre Ausdrücke heißen:
- e) Das Verfahren zum Graphendurchlauf, das dem Level-Order-Durchlauf bei einem Baum entspricht, heißt:
- f) Wie heißen in einem Baum Knoten, die keine Kinder besitzen?
- g) Welche Laufzeit hat folgender Algorithmus? Geben Sie die Zeit in der O(n)-Notation an.  

```
for (int i=0; i<n; i++) {
    for (int j=0; j<n; j++) {
        k++;
    }
}
```
- h) Wie nennt man einen Baum, bei dem alle Ebenen komplett besetzt sind?
- i) Nennen Sie zwei in der Vorlesung behandelte Entwurfsprinzipien.
- j) Gegeben sei der reguläre Ausdruck

$$((A \mid B) \mid CB)^* (A \mid C)$$

Welche der folgenden Buchstabenfolgen erfüllen den regulären Ausdruck?

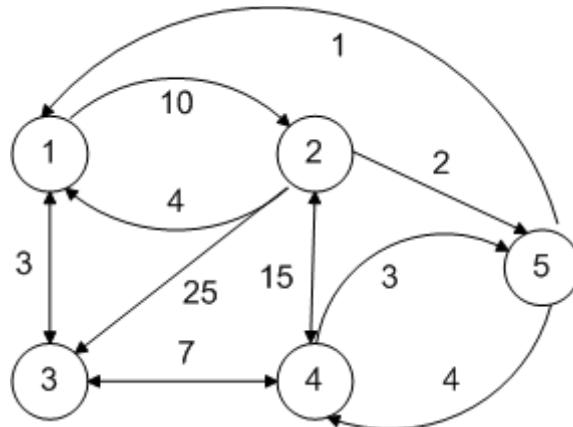
ABC, ACAC, BCBBBAC, C, ACB

- k) Welches Sortierverfahren verbirgt sich hinter dem folgenden Codeschnipsel?

```
//Codeschnipsel
if (array[j]>m) {
    array[j+1] = array[j];
} else {
    break;
}
```

## Aufgabe 2

- Stellen Sie die Adjazenzmatrix für den gegebenen Graphen auf.
- Finden Sie mithilfe des Dijkstra-Algorithmus die minimalen Kosten, um von Knoten 2 zu jedem anderen Knoten des Graphen zu gelangen.



## Aufgabe 3

Sortieren Sie die unten angegebene Zahlenfolge in aufsteigender Reihenfolge. Benutzen Sie den Quicksort-Algorithmus. Wählen Sie das rechte Element einer Teilliste als Pivot-Element. Machen Sie die Pivot-Elemente, die Teilfelder, sowie jeden Tauschvorgang kenntlich. Für Teilfelder von 2 Elementen brauchen Sie den Quicksort-Algorithmus nicht mehr zu verwenden, sondern dürfen die Elemente gegebenenfalls einfach umtauschen.

7	9	11	6	3	2	1	5	12	4	8
---	---	----	---	---	---	---	---	----	---	---

## Aufgabe 4

Gegeben sei das folgende Feld:

20	16	21	3	6	11	42
----	----	----	---	---	----	----

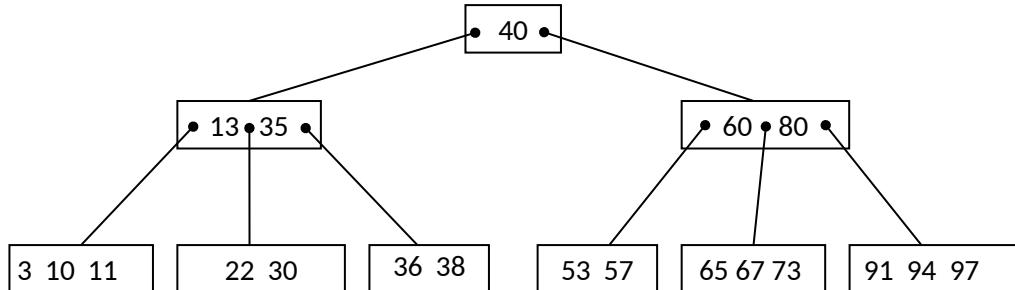
Sortieren Sie das Feld mit Heap-Sort.

- a) Wenden Sie einen der in der Vorlesung vorgestellten Algorithmen an, um das Feld in einen Heap zu verwandeln. Geben Sie jeweils nach dem Einfügen einer Zahl den entstandenen Heap an, wahlweise als Feld (Tabellenvorlage) oder als Baum.

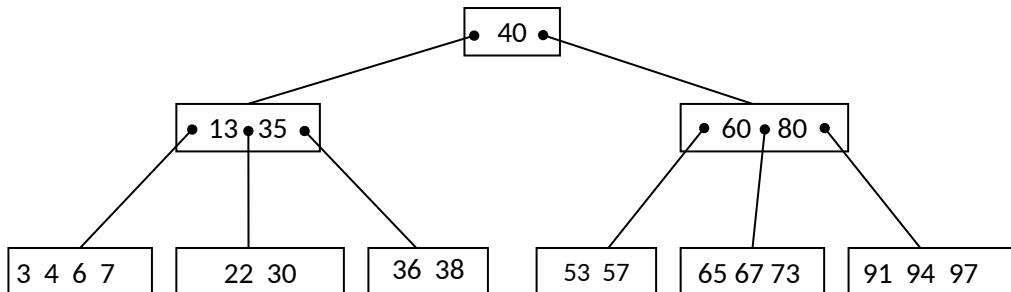

- b) Sortieren Sie die Elemente durch Auflösung des Heaps. Geben Sie für jeden Zwischenschritt das aktuelle Feld an (einschließlich der bereits sortierten Elemente). Tipp: Sie können sich zusätzlich als Hilfestellung auch jeweils den Heap in Baumform notieren.


## Aufgabe 5

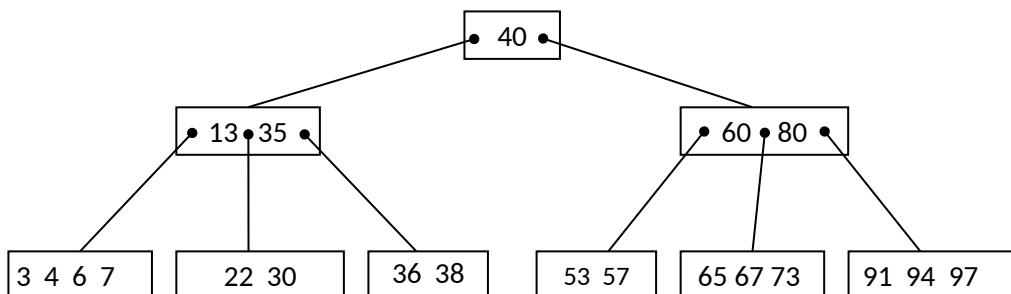
- a) Gegeben sei folgender B-Baum der Ordnung 2. Löschen Sie die **38**.



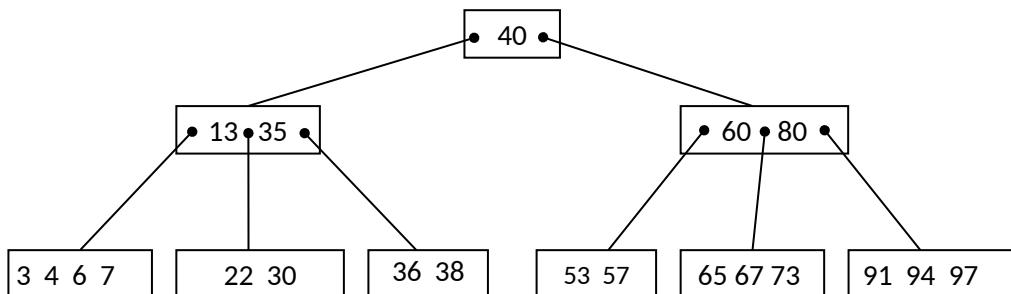
- b) Gegeben sei folgender B-Baum der Ordnung 2. Löschen Sie die **22**.



- c) Gegeben sei folgender B-Baum der Ordnung 2. Löschen Sie die **40**.



- d) Gegeben sei folgender B-Baum der Ordnung 2. Fügen Sie die **5** ein.



## Aufgabe 6

Gegeben sei der folgende Code:

```
public static boolean analyzeString(String x) {  
    for (int i=0; i<x.length()-1; i++) {  
        if (x.substring(i+1).contains(""+x.charAt(i))) {  
            return false;  
        }  
    }  
    return true;  
}
```

- a) Was prüft der angegebene Code?
- b) Welche Laufzeitkomplexität hat der Code in Abhängigkeit von der Länge des Strings x? Geben Sie den Worst-Case an. Begründen Sie Ihre Antwort. Tipp: In der substring-Methode wird der Teilstring in den neuen String kopiert.
- c) Skizzieren Sie in Worten einen Algorithmus, der die Laufzeitkomplexität im Worst Case gegenüber dem angegebenen Code verbessert. Nennen Sie auch die neue Laufzeitkomplexität.

## Aufgabe 7

Implementieren Sie die Methode

```
public static ArrayList<Integer> textSearch(String text, String pattern),
```

die ein übergebenes Pattern in einem übergebenen Text sucht und ein Array zurückgibt, in dem die Startindizes aller Vorkommen des Patterns im Text gespeichert sind.

Das Pattern kann neben Zahlen und Groß- und Kleinbuchstaben auch das Zeichen ,?’ enthalten, das als Platzhalter für ein beliebiges Zeichen steht.

## Aufgabe 8

Ein Knoten eines Binärbaums habe folgendes Aussehen:

```
public class Node {
    public String cont;
    public Node left;
    public Node right;
}
```

Schreiben Sie eine Funktion

```
public static boolean isComplete(Node root),
```

die überprüft, ob der Baum, dessen Wurzel der übergebene Knoten ist, vollständig ist oder nicht. Ist der Baum vollständig (alle Ebenen sind komplett besetzt), wird `true` zurückgegeben, ansonsten `false`.

## Aufgabe 9

- a) Implementieren Sie eine Klasse `Kind` mit folgenden Attributnen:

```
String name;
int alter;
und einem passenden Konstruktor.
```



- b) Implementieren Sie eine Klasse `Ringelreihen`, die eine Gruppe von Kindern kreisförmig anordnet (siehe Bild). Erweitern Sie eventuell die Klasse `Kind`. Die Klasse `Ringelreihen` soll einen Zeiger auf ein Kind des Kreises besitzen. Ferner soll die Klasse folgende Methoden enthalten:

```
public Ringelreihen(Kind k1, Kind k2) /*Erzeugt ein Ringelreihen mit zwei Kindern*/
```

```
public void moveLeft() /*verschiebt den Zeiger um ein Kind gegen die Uhrzeigerrichtung*/
public void moveRight() /*verschiebt den Zeiger um ein Kind in Uhrzeigerrichtung*/
public void insert(Kind k) /*fügt ein neues Kind in den Reigen (links vom aktuellen Kind) ein und setzt den Zeiger auf das neue Kind.*/
public int size() /*Gibt die Anzahl der Kinder im Reigen zurück.*/
public void setToYoungest() /*Setzt den Zeiger auf das jüngste Kind oder auf eines der jüngsten, falls es mehrere jüngste Kinder gibt.*/
```

Benutzen Sie keine vorgefertigten Java-Klassen, wie z.B. `ArrayList` oder `LinkedList`.