

Beantworten Sie die Fragen in den Aufgaben 1 und 2 mit einer kurzen, prägnanten Antwort.

Aufgabe 1 (8 Punkte)

1. Gegeben sei folgender Ausdruck $(a \wedge b) \ \&\& \ (a \ || \ b) \ \&\& \ a$. Welchen Wert hat er für $a = \text{true}$ und $b = \text{false}$.

2. Ein Programm sei *partiell korrekt*. Was muss noch gelten, damit das Programm *total korrekt* ist?

3. Wie ist die untere Schranke der Laufzeit eines Sortieralgorithmus, der auf Vergleichen von n Elementen beruht, in der O-Notation?

4. Wie lautet die Laufzeitklasse in O-Notation von Quicksort im *Worst case*?

5. Was ist ein *abstrakter Datentyp* (ADT)?

6. Nach welchem Prinzip arbeitet eine Schlange?

7. Welcher Graphen-Algorithmus aus der Vorlesung löst das *all-pairs-shortest-path*-Problem?

8. Wie nennt man einen Rundweg in einem Graphen, der jeden Knoten genau einmal besucht und beim Ausgangsknoten wieder endet?

Aufgabe 2 (8 Punkte)

Alle Fragen beziehen sich auf die Programmiersprache Java ab Version 5.

1. Wie nennt man den Vorgang, einem Objekt zur Laufzeit seinen Programmcode zuzuordnen?

2. Wieviele Bits werden für die Codierung einer `int`-Zahl verwendet?

3. Von welcher Klasse erben alle Javaklassen implizit?

4. Wie nennt man Methoden, wenn sie in einer Klasse den gleichen Namen aber verschiedene Parameter haben?

5. Wie viele Interfaces kann eine Klasse implementieren?

6. Mit welchem Aufruf vergleicht man den Inhalt von zwei Strings `s` und `t` lexikografisch?

7. Was bedeutet das Schlüsselwort `private` bei Methoden in Java?

8. Wie lautet der Javacode, um eine typsichere `HashMap` `map` zu deklarieren und zu instanziiieren, die Schlüssel vom Typ `Integer` und Werte vom Typ `Person` aufnehmen kann?

Tragen Sie bei den Aufgaben 3 bis 6 Ihre Lösungen in den vorgesehenen Platz ein.

Aufgabe 3 (10 Punkte)

a) Tragen Sie die 32-Bit-Codierung nach Definition des Skriptes für die Zahl 13.6 in die vorgesehenen Kästchen ein (3 Punkte).

Vorzeichen	Exponent	reduzierte Mantisse

b) Welche Zahl wird durch folgende Bitfolge codiert (3 Punkte)? Geben Sie Zwischenschritte und das Ergebnis als Dezimalzahl an.

1 11111110 100000000000000000000000

c) Gegeben zwei 6-Bit-Dualzahlen im Zweierkomplement:

111001

101010

Ziehen Sie die zweite Zahl von der ersten ab und geben Sie das Ergebnis sowohl im 6-Bit-Zweierkomplement als auch als Dezimalzahl an. Überprüfen Sie die Rechnung nach dem aus der Vorlesung bekannten Verfahren auf Korrektheit. Schreiben Sie zu Ihren Rechenschritten **stichwortartig** auf, was Sie tun. (4 Punkte)

Aufgabe 4 (6 Punkte)

Gegeben sei folgende Java-Klasse:

```
public class Foo {
    public static int bar(int n) {

        int ergebnis = 0;

        for (int j = 1; j <=n; j++) {
            for (int i = 1; i < 2*n; i+=2) {
                ergebnis = ergebnis + i;
            }
        }

        return ergebnis;
    }
}
```

a) Was berechnet die Methode `public static int bar(int n)` der Klasse `Foo`? (4 Punkte)

b) Wie ist die Laufzeit der Methode `public static int bar(int n)` in der O-Notation für $n \geq 0$? (2 Punkte)

Aufgabe 5 (4 Punkte)

Sortieren Sie die Zahlenfolge

7 5 3 6 4 1 8 2

mit dem Selectionsort-Verfahren. Stellen Sie die Arbeitsweise des Algorithmus mit geeigneten Zwischenschritten da.

Aufgabe 6 (6 Punkte)

Gegeben sei folgendes `int`-Array:

2 4 8 1 6 5

Sortieren Sie das `int`-Array mit dem Heapsort-Verfahren aus der Vorlesung.

a) Geben Sie für die erste Phase den Baum sowie den daraus erzeugten Heap an.
(2 Punkte)

Ursprungsbaum:

Heap:

b) Für Phase 2 geben Sie jeweils den Baum nach Entfernen der Wurzel, den reorganisierten Heap sowie das zugehörige aktuelle `int`-Array für jeden Sortierschritt nach dem Reorganisieren an. (4 Punkte)

Bei den Aufgaben 7 bis 9 müssen Sie selber etwas zeichnen. Nutzen Sie den dafür vorgesehenen Platz.

Aufgabe 7 (8 Punkte)

Der Verein der Osnabrücker Spaß- und Faschings-Freunde (SUFF) hat den Osnabrücker Kamelle-Austeil-Automaten (OSKAA) entwickelt. Dieser muss befüllt, gespannt und ausgelöst werden.

- Vor dem Auslösen muss der Automat mindestens einmal gespannt werden, damit er Kamelle verteilt.
- Wird der Automat in ungespanntem Zustand ausgelöst, passiert nichts.
- Man kann den Automaten beliebig oft spannen.
- Aus Sicherheitsgründen darf der Automat nicht im leeren, gespannten Zustand ausgelöst werden.
- Nach dem Befüllen kann der Automat maximal zweimal Kamelle verteilen, bevor er neu befüllt werden muss.
- Ein komplett voller Automat läuft über, wenn man ihn nochmals befüllt, daher darf er dann nicht befüllt werden.

a) Geben Sie das Eingabealphabet und die Zustandsmenge von OSKAA an. Wenn Sie das Eingabealphabet mit Abkürzungen angeben, geben Sie an, wofür die Abkürzung steht. Falls Sie die Zustände nummerieren, geben Sie eine kurze Beschreibung des jeweiligen Zustands an. (2 Punkte)

Eingabealphabet:	
Zustandsmenge:	

b) Zeichnen Sie den Zustandsüberföhrungsgraphen des endlichen Automaten, anhand dessen man erkennen kann, ob sich OSKAA in einem Zustand befindet, in dem er zum Kamelleverteilen ausgelöst werden kann. Markieren Sie den Startzustand und den oder die korrekten Endzustand/Endzustände. (6 Punkte)

Aufgabe 8 (6 Punkte)

Gegeben seien die Preorder- und die Inorder-Traversierung eines binären Baumes:

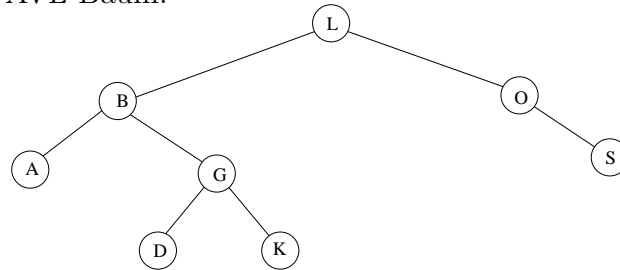
Preorder: 16 19 13 42 21 23 18 25 11

Inorder: 19 13 16 23 18 21 42 25 11

Zeichnen Sie den zugehörigen Baum.

Aufgabe 9 (8 Punkte)

Gegeben sei folgender AVL-Baum:



a) Geben Sie die Inorder-Traversierung und die Tiefensuche des Baumes an. (2 Punkte)

Inorder	
Tiefensuche	

b) Fügen Sie in den Baum den Buchstaben Z ein, indem Sie den Knoten direkt in den obigen AVL-Baum einzeichnen. Versehen Sie den kompletten Baum mit Balancen. (1 Punkt)
 Führen Sie eine eventuell notwendige Rotation aus und zeichnen Sie den reorganisierten Baum erneut. Vermerken Sie, welche Rotation Sie verwenden haben. (2 Punkte)

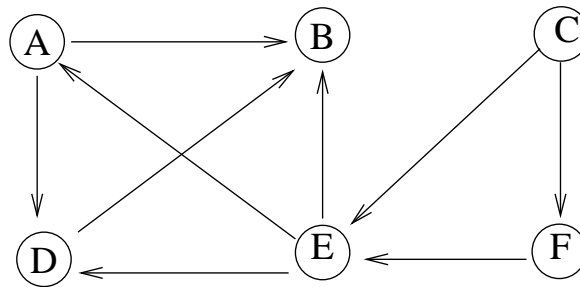
c) Fügen Sie in den neu gezeichneten Baum in Aufgabenteil b) den Buchstaben E ein und versehen Sie den kompletten Baum mit Balancen. (1 Punkt)
 Führen Sie die notwendige Rotation aus und zeichnen Sie den reorganisierten Baum unten nochmals. Vermerken Sie die Art der Rotation, die Sie verwendet haben (2 Punkte).

Tragen Sie bei den Aufgaben 10 und 11 Ihre Lösungen in den vorgesehenen Platz ein.

Aufgabe 10 (6 Punkte)

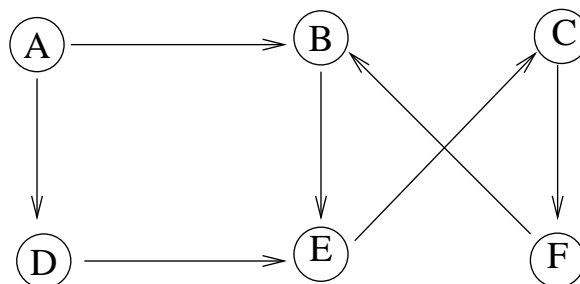
Geben Sie jeweils an, ob der Graph topologisch sortiert werden kann. Ist der Graph topologisch sortierbar, geben Sie die zugehörige Sortierung an. Wenn der Graph nicht topologisch sortierbar ist, begründen Sie, warum.

a)



topologisch sortierbar?	
Sortierung oder Begründung	

b)



topologisch sortierbar?	
Sortierung oder Begründung	

Aufgabe 11 (7 Punkte)

Gegeben sei die Hashfunktion h , die eine positive ganze Zahl auf ihre einstellige Quersumme abbildet.

Beispiele: $h(22) = 4$, $h(87) = 6$

Außerdem ist folgende Hashtabelle gegeben:

$h(x)$	Element x
1	9
2	109
3	
4	
5	14
6	50
7	
8	73
9	32

a) Geben Sie die Art des Hashings an, mit der diese Hashtabelle erzeugt wurde. (2 Punkte)

b) Geben Sie die Reihenfolge an, in der die Elemente eingefügt wurden. (3 Punkte)

c) Geben Sie den Hashwert für die Zahl 89 an und fügen Sie sie in die Tabelle oben ein. (2 Punkte)

Aufgabe 13 (5 Punkte)

Für nichtnegative ganze Zahlen n und k ist der Binomialkoeffizient wie folgt definiert:

$$\binom{n}{k} = 0 \text{ für } n < k,$$

$$\binom{n}{k} = \frac{\binom{n}{k-1} \cdot (n-k+1)}{k} \text{ und } \binom{n}{0} = 1 \text{ für } n \geq k$$

Ergänzen Sie in der Klasse `BinomialkoeffizientRekursiv` die Methode `binKoeff(int n, int k)`, die den Binomialkoeffizienten **rekursiv** berechnet. Achten Sie auf eine Fehlerbehandlung und werfen Sie dabei eine `RuntimeException`.

```
public class BinomialkoeffizientRekursiv{  
  
    public static long binKoeff(int n, int k){
```

```
    }  
}
```

Aufgabe 14 (5 Punkte)

Wie in Aufgabe 13 ist für nichtnegative ganze Zahlen n und k der Binomialkoeffizient wie folgt definiert:

$$\binom{n}{k} = 0 \text{ für } n < k,$$

$$\binom{n}{k} = \frac{\binom{n}{k-1} \cdot (n-k+1)}{k} \text{ und } \binom{n}{0} = 1 \text{ für } n \geq k$$

Ergänzen Sie in der Klasse `BinomialkoeffizientIterativ` die Methode `binKoeff(int n, int k)`, die den Binomialkoeffizienten **iterativ** berechnet. Führen Sie die Fehlerbehandlung analog zu Aufgabe 13 durch.

Hinweis: Überlegen Sie sich zur iterativen Berechnung des Binomialkoeffizienten, wie die Formel

$$\binom{n}{k} = \binom{n}{k-1} \cdot \frac{n-(k-1)}{k}$$

ausmultipliziert aussieht.

```
public class BinomialkoeffizientIterativ{

    public static long binKoeff(int n, int k){

        }

    }
```

