

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

Aufgabe 1 (8 Punkte)

1. Wie sieht 6 in der 6Bit 2-er Komplementdarstellung aus?

2. Welche Laufzeit in O-Notation hat jeder Sortieralgorithmus, der auf Vergleichen beruht, mindestens?

3. Wie nennt man einen für einen Compiler formulierten Algorithmus?

4. Welche Methode hat jeder der in der Vorlesung behandelten ADTs?

5. Nach welchem Prinzip arbeitet der rekursive Mergsort?

6. Wie viele Ebenen hat ein Heap mit 12 Knoten?

7. Wie viele Knoten hat ein binärer Baum der Höhe h minimal?

8. Wie nennt man einen Weg, der jeden Knoten eines gerichteten Graphen genau einmal besucht und am Ausgangsknoten wieder endet?

Aufgabe 2 (8 Punkte)

Alle Fragen beziehen sich auf die Programmiersprache Java 5.

1. Was bedeutet `super()` innerhalb eines Konstruktors?

2. Sei `int i = 9;` und `int j = 4;`. Welchen Wert hat der Ausdruck `(i + 1) / j`?

3. Wie viel Bit werden für die Codierung einer `long`-Zahl verwendet?

4. Welche Ausgabe hat die Anweisung `IO.println(42 < 23 ? 42 : 19);`?

5. Wie viele Konstruktoren hat eine Klasse mindestens?

6. Wie nennt man Methoden, die in der gleichen Klasse den selben Namen aber unterschiedliche Parameter haben?

7. Welchen `boolean`-Wert liefert der Ausdruck `true ^ true`?

8. Von wie vielen Klassen kann eine Klasse maximal erben?

Tragen Sie bei den Aufgaben 3 bis 10 Ihre Lösungen in den vorgesehenen Platz ein.

Aufgabe 3 (6 Punkte)

Gegeben sei folgende Java-Klasse:

```
import AlgoTools.IO;

public class Aehnlich {

    private int a, b;

    public Aehnlich(int a, int b) {
        this.a = a;
        this.b = b;
    }
    public void aehnlich(int a) {
        this.b += a;
        aehnlich();
    }

    public void aehnlich(int a, int b){
        this.a += b;
        this.b += a;
        aehnlich(this.b);
    }
    public void aehnlich(){
        a *= 2;
        b *= 2;
    }
    public static void main(String[] args) {

        Aehnlich aehnlich1 = new Aehnlich(4,1);
        Aehnlich aehnlich2 = new Aehnlich(3,2);

        aehnlich1.ahnlich(aehnlich1.a, aehnlich1.b);
        aehnlich2.ahnlich(aehnlich2.a);

        IO.println(aehnlich1.a + aehnlich1.b + aehnlich2.a + aehnlich2.b);
    }
}
```

Welche Ausgabe hat die Methode main dieser Klasse?

Aufgabe 4 (6 Punkte)

Gegeben sei folgende Java-Klasse:

```
import AlgoTools.IO;

public class Fraglich {

    public static void fraglich(int n) {

        Keller k = new VerweisKeller();

        while (n > 0) {
            k.push(n % 8);
            n /= 8;
        }

        while (!k.empty()) {
            IO.print(k.top());
            k.pop();
        }
    }

    public static void main(String[] argv) {

        fraglich(123);
    }
}
```

a) Was berechnet die Methode `public static void fraglich(int n)` der Klasse `Fraglich` für $n > 0$? (3 Punkte)

b) Welche Ausgabe hat der Aufruf `fraglich(123)` in der `main`-Methode? (1 Punkt)

c) Geben Sie in der O-Notation die Laufzeit in Abhängigkeit vom Wert der Eingabe $n > 0$ an. (2 Punkte)

Aufgabe 7 (4 Punkte)

Nennen Sie für die Sortierverfahren in der Tabelle jeweils die Komplexitätsklasse in der O-Notation. Legen Sie jeweils die beste Implementation des Sortieralgorithmus zugrunde.

	best case	average case	worst case
MergeSort			
BubbleSort			
SelectionSort			
QuickSort			

Aufgabe 8 (6 Punkte)

Sie analysieren eine Methode und erstellen eine Rekursionsgleichung mit einer Funktion $f(n)$, die das Laufzeitverhalten der Methode charakterisiert. Tragen Sie an den entsprechenden Stellen die Laufzeit in der O-Notation ein.

Hinweis: Unter Umständen kann es hilfreich sein, zunächst eine Wertetabelle anzulegen.

$$f(n) = \begin{cases} 0, & \text{falls } n = 0; n = 1 \\ n + f(\frac{n}{2}) & \text{sonst} \end{cases}$$

Laufzeit in O-Notation:

$$f(n) = \begin{cases} 0, & \text{falls } n = 0; n = 1 \\ 2n - 1 + f(n - 1) & \text{sonst} \end{cases}$$

Laufzeit in O-Notation:

Bei den Aufgaben 10 bis 13 müssen Sie selber etwas zeichnen. Nutzen Sie den dafür vorgesehenen Platz.

Aufgabe 10 (6 Punkte)

Sie befüllen eine Waschmaschine, um Wäsche zu waschen. Dazu können Sie beliebig oft, müssen aber mindestens einmal Wäsche in die Waschtrommel tun. Vorher, nachher oder auch zwischendurch müssen Sie genau einmal Waschpulver in die Maschine einfüllen. Als letztes stellen Sie die Maschine an.

Erstellen Sie einen endlichen Automaten, der den Ablauf der Waschmaschinenbedienung modelliert. Zeichnen Sie dazu den Zustandsüberföhrungsgraphen des Automaten. Markieren Sie den Start- und den oder die korrekten Endzustände. Geben Sie auch das Eingabealphabet an.

Aufgabe 11 (6 Punkte)

Gegeben sei der folgende Postfix-Ausdruck:

2 3 * 9 5 % + 4 6 3 / - /

a) Zeichnen Sie den zugehörigen Baum mit Operatoren in den inneren Knoten und Operanden in den Blättern. (4 Punkte)

b) Wie lautet die Preorder-Traversierung dieses Baums? (1 Punkt)

c) Berechnen Sie das Ergebnis und geben Sie den numerischen Wert an. (1 Punkt)

Aufgabe 12 (6 Punkte)

a) Fügen Sie nacheinander folgende Zahlen in einen Suchbaum ein:

8 18 16 1 6 10 13 3

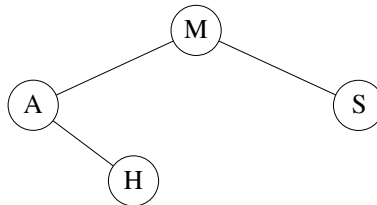
Zeichnen Sie den Baum nach dem Einfügen der 3. (4 Punkte)

b) Löschen Sie aus dem Baum die Zahl 18 und zeichnen Sie den Baum erneut. (1 Punkt)

c) Löschen Sie aus dem Baum die Zahl 8 und zeichnen Sie den Baum erneut. (1 Punkt)

Aufgabe 13 (6 Punkte)

Gegeben sei folgender AVL-Baum:



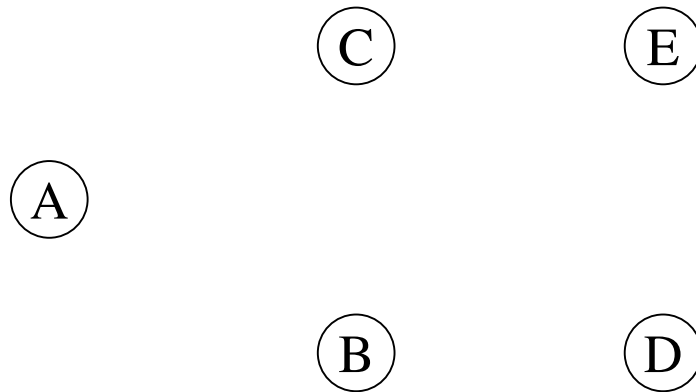
- a) Fügen Sie in den Baum den Buchstaben **F** ein, indem Sie den Knoten direkt in den obigen AVL-Baum einzeichnen. Versehen Sie den **kompletten** Baum mit Balancen. (1 Punkt)
- b) Führen Sie eine eventuell notwendige Rotation aus und zeichnen Sie den reorganisierten Baum erneut. Vermerken Sie, welche Rotation Sie verwenden haben. (2 Punkte)
- c) Fügen Sie in den neu gezeichneten Baum in Aufgabenteil b) den Buchstaben **D** ein und versehen Sie den **kompletten** Baum mit Balancen (1 Punkt).
- d) 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).

Aufgabe 14 (6 Punkte)

Sei für einen gerichteten, bewerteten Graph G folgende Adjazenzmatrix gegeben:

		nach				
		A	B	C	D	E
von	A	0	2	1	∞	∞
	B	∞	0	2	1	∞
	C	4	∞	0	3	5
	D	∞	2	2	0	1
	E	∞	∞	∞	3	0

a) Zeichnen Sie den durch die Adjazenzmatrix erzeugten Graphen G . (4 Punkte)



b) Welches ist der kürzeste Weg von Knoten A nach Knoten E bzgl. der Kantenbewertung? (1 Punkt)

c) Welchen Wert hat dieser kürzeste Weg? (1 Punkt)

Aufgabe 17 (8 Punkte)

Ein winziger nachtaktiver Käfer beschloss eines Abends, einen Baum hochzuklettern. Der Baum war zu diesem Zeitpunkt genau 2m hoch. In jeder Nacht schaffte der Käfer eine Strecke von 10cm. Der Baum wuchs aber an jedem Tag gleichmäßig 20cm entlang seiner gesamten Länge.

Vervollständigen Sie die Methode `dauer` der Klasse `BaumKaefer`, so dass sie die Anzahl der Tage berechnet, die der Käfer braucht, um an der Baumspitze anzukommen. Beachten Sie dabei, dass jeder Tag abends beginnt, d.h. an einem Tag krabbelt zunächst der Käfer 10cm den Baum hinauf, anschließend wächst der Baum um 20cm.

```
public class BaumKaefer {  
  
    public static int dauer() {
```

```
    }  
}
```