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

Aufgabe 1 (8 Punkte)

1. Wieviele verschiedene positive Zahlen (ohne 0) können im 4-Bit-Zweierkomplement dargestellt werden?

2. Nennen Sie ein Sortierverfahren, das nach dem Prinzip *Divide & Conquer* arbeitet.

3. Sie fügen die Zahlen `4 2 3 1` in einen leeren Suchbaum ein. Welche Zahl steht anschließend in der Wurzel?

4. Welche Methoden stellt das Interface `Menge` zur Verfügung?

5. Wieviele Elemente sind in einem *Heap* der Höhe 4 mindestens enthalten?

6. Welches Verfahren wird zum Nachweis der *partiellen Korrektheit* verwendet?

7. Welcher Algorithmus aus der Vorlesung berechnet in einem Graphen den kürzesten Weg zwischen zwei vorgegebenen Knoten?

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

Aufgabe 2 (8 Punkte)

Alle Fragen beziehen sich auf die Programmiersprache Java 5.

1. Was bedeutet `super` innerhalb einer Methode?

2. Mit welchem Audruck erzeugen Sie ein `Integer`-Array der Länge 10?

3. Mit welchem Schlüsselwort kennzeichnet man Variablen, denen man nur einmal einen Wert zuweisen darf?

4. Welche Ausgabe hat die Anweisung `IO.println("1 < 2 ? 3 : 4");`?

5. Wie lautet der Methodenkopf der Klassenmethode `methode` ohne Rückgabewert, die einen `int`-Wert übergeben bekommt und nur in derselben Klasse sichtbar ist?

6. Welche unbehandelten `Exception`s müssen nicht mit einer `throws`-Klausel angekündigt werden?

7. Mit welcher Anweisung weisen Sie der Variablen `float f` den Wert 17,5 zu?

8. Welchen Datentyp verwenden Sie, wenn Sie Schlüssel-Wert-Paare speichern wollen?

Tragen Sie bei den Aufgaben 3 bis 8 Ihre Lösungen in den vorgesehenen Platz ein.

Aufgabe 3 (5 Punkte)

Gegeben sei folgende Java-Klasse:

```
public class Fraglich1 {

  public static void fraglich(int a, int b) {

    int n = 0;

    if (b < 0) {
      a = -a;
      b = -b;
    }

    while (b > 0) {

      if (b % 2 == 1)
        n += a;

      a += a;
      b /= 2;
    }

    return n;
  }
}
```

a) Was berechnet die Methode `public static void fraglich(int a, int b)` der Klasse `Fraglich1`? (4 Punkte)

b) Geben Sie die Komplexitätsklasse der Laufzeit in der O-Notation an. (1 Punkt)

Aufgabe 4 (7 Punkte)

Gegeben sei folgende Java-Klasse:

```
import AlgoTools.IO;

public class Fraglich2 {

  public static void fraglich(int n) {

    if (n < 2)
      throw new RuntimeException("n zu klein!");

    int i = 2;

    while (i <= n) {
      if ((n % i) == 0) {
        n /= i;
        IO.print(i + " ");
      }
      else
        i++;
    }
  }
}
```

a) Was gibt die Methode `public static void fraglich(int n)` der Klasse `Fraglich2` für `n` ≥ 2 aus? (3 Punkte)

b) Geben Sie die Komplexitätsklasse der Laufzeit in der O-Notation für den *best* und *worst case* an. (2 Punkte)

best case	
worst case	

c) Geben Sie die beiden Mengen an, für die der jeweilige Fall eintritt. (2 Punkte)

best case	
worst case	

Aufgabe 5 (3 Punkte)

Wie lautet die Dezimalzahldarstellung der Fließkommazahl, die durch folgende Bitfolge kodiert ist? Geben Sie auch die Einzelwerte von Vorzeichen, Exponent und Mantisse an.

```
1   1 1 1 1 1 1 1 1   1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Vorzeichen	
Exponent	
Mantisse	
Ergebnis	

Aufgabe 6 (6 Punkte)

Gegeben seien zwei Integer-Arrays A und B mit der Länge n. A sei aufsteigend, B sei absteigend sortiert. Geben Sie die Komplexitätsklasse der Laufzeit in der O-Notation an, die die folgenden Sortierverfahren benötigen, wenn sie auf die beiden Arrays A und B angewandt werden.

Sortierverfahren	Array A	Array B
Mergesort		
Bubblesort		
Quicksort		

Aufgabe 7 (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 dar.

Aufgabe 8 (6 Punkte)

Gegeben seien folgende Hashfunktionen:

$$f_1(x) = 2x \bmod 11$$
$$f_2(x) = (x^2 + 2) \bmod 7$$

Fügen Sie die Zahlen 14, 9, 3, 7, 2 und 0 der Reihenfolge nach gemäß der Hashfunktion f_1 in die folgende Hashtabelle ein. Gehen Sie nach dem Verfahren des geschlossenen Hashings vor und verwenden Sie als Sondierungsverfahren Double Hashing mit der Hashfunktion f_2.

$f_1(x)$	Element x
0	11
1	
2	1
3	
4	13
5	
6	
7	
8	
9	
10	

Bei den Aufgaben 9 bis 13 müssen Sie selber etwas zeichnen. Nutzen Sie den dafür vorgesehenen Platz.

Aufgabe 9 (8 Punkte)

Erstellen Sie einen Endlichen Automaten, der durch 4 teilbare Dualzahlen erkennen soll. Dabei erhält der Automat sukzessive die Bits der Dualzahl beginnend von links. Jedes neue Bit stellt eine Eingabe in den Automaten dar.

Geben Sie die Zustandsmenge des Automaten mit Bedeutung der einzelnen Zustände an. Zeichnen Sie den Zustandsüberführungsgraphen des Automaten. Markieren Sie den Start- und den oder die korrekten Endzustände.

Aufgabe 10 (5 Punkte)

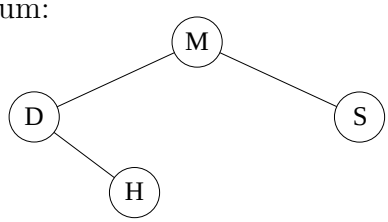
Gegeben sei die Postorder-Traversierung eines Suchbaums:

```
8  25  32  33  31  27  35  36  41  34
```

Zeichnen Sie den zugehörigen Baum.

Aufgabe 11 (6 Punkte)

Gegeben sei folgender AVL-Baum:



a) Fügen Sie in den Baum die Buchstaben **B** und **K** ein, indem Sie die 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 verwendet haben. (2 Punkte)

c) Fügen Sie in den neu gezeichneten Baum in Aufgabenteil b) den Buchstaben **A** ein und versehen Sie den **kompletten** Baum mit Balancen. (1 Punkt)

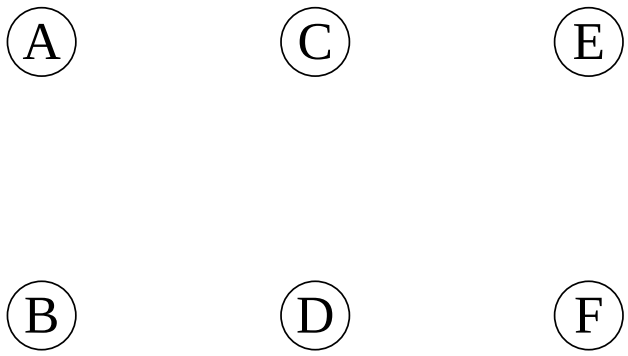
d) Führen Sie die notwendige Rotation aus und zeichnen Sie den reorganisierten Baum unten nochmals. Vermerken Sie die Art der Rotation, die Sie verwendet haben. (2 Punkte)

Aufgabe 12 (8 Punkte)

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

		nach					
		A	B	C	D	E	F
von	A	0	2	6	∞	∞	∞
	B	∞	0	3	7	∞	∞
	C	∞	∞	0	1	8	∞
	D	4	8	9	0	2	2
	E	2	∞	2	∞	0	∞
	F	∞	3	7	∞	5	0

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

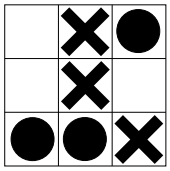


b) Geben Sie die Knoten eines möglichen *Hamiltonkreises* in der entsprechenden Reihenfolge an. (2 Punkte)

c) Welches ist der kürzeste Weg von Knoten A nach Knoten E bzgl. der Kantenbewertung? (2 Punkte)

Aufgabe 13 (7 Punkte)

Betrachten Sie folgende Situation eines TicTacToe-Spiels auf einem 3×3-Feld:



Ihre Spielzüge sind mit einem Kreis, die Ihres Gegners mit $\times$ eingezeichnet. Sie sind am Zug.

a) Zeichnen Sie einen Spielbaum für den restlichen Verlauf dieses TicTacToe-Spiels. Bewerten Sie die möglichen Spielzüge und schreiben Sie die entsprechenden *minmax*-Werte für jeden Spielzug auf. Verwenden Sie die Werte `-1` (Kreuz/Gegner gewinnt), `0` (unentschieden) und `+1` (Kreis/man selbst gewinnt). (6 Punkte)

b) Wie wird das Spiel ausgehen, wenn sowohl Ihr Gegner als auch Sie möglichst geschickt spielen? (1 Punkt)

Bei den Aufgaben 14 bis 16 müssen Sie selbst einige Zeilen Java-Code schreiben. Nutzen Sie den vorgesehenen Platz zwischen den geschweiften Klammern.

Aufgabe 14 (4 Punkte)

Sei folgende Gleichung gegeben:

$$(n+1)^2 = n^2 + 2n + 1 \qquad \forall n \in \mathbb{N}$$

Implementieren Sie die **rekursive** Methode `public static int square(int n)`, die mit Hilfe dieser Formel das Quadrat einer natürlichen Zahl bestimmt. Verwenden Sie keine Schleifen.

```
public class Funktion {

  public static int square(int n) {









  }
}
```

Aufgabe 15 (8 Punkte)

Sei folgendes Interface gegeben:

```
public interface Queue {
  public void enqueue(Object o);
  public Object dequeue();
  public boolean empty();
}
```

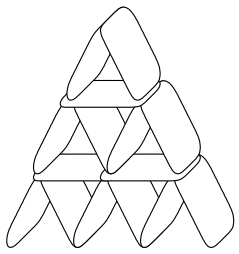
Das Interface definiert den ADT Queue, der dem ADT Schlange ähnelt, mit dem Unterschied, dass die Methode `dequeue` das vorderste Element nach dem Löschen direkt zurückliefert und die Methode `front` fehlt.

Sei `QueueImpl` eine Klasse, die das Interface `Queue` implementiert. Implementieren Sie die von `QueueImpl` abgeleitete Klasse `FrontQueue`, indem Sie die fehlende Methode `public Object front()` ergänzen. Die Methode liefert das vorderste Element der Queue zurück. Nach Beendigung der Methode ist die Queue unverändert. Verwenden Sie nur die Operationen des ADT Queue und implementieren Sie keine weiteren Methoden.

```
public class FrontQueue extends QueueImpl {

  public Object front() {









  }
}
```

Aufgabe 16 (7 Punkte)

Ein Kartenhaus aus Spielkarten besteht aus mindestens einer Etage. Die oberste Etage, das sog. Dach, bilden zwei Karten. Die `n`-te Etage von oben besteht aus `n` Dächern und `n-1` eingezogenen Decken (horizontalen Karten).



Kartenhaus der Hoehe 3

Die Methode `anzahlRekursiv` der unten gegebenen Klasse `Kartenhaus` liefert die Anzahl der benötigten Karten in Abhängigkeit von der Höhe des Kartenhauses, angegeben in Etagen.

a) Ergänzen Sie die Methode `anzahlIterativ` der Klasse `Kartenhaus`, so dass sie ebenfalls die Anzahl der benötigten Karten für das Kartenhaus liefert, allerdings ohne Rekursion zu benutzen und ohne die Methode `anzahlRekursiv` aufzurufen. (5 Punkte)

```
import AlgoTools.IO;
public class Kartenhaus {
  public static void main(String[] argv) {
    IO.println(anzahlRekursiv(4) + anzahlRekursiv(2));
  }
  public static int anzahlRekursiv(int hoehe) {
    if (hoehe <= 0)
      return 0;
    return 2 * hoehe + hoehe - 1 + anzahlRekursiv(hoehe - 1);
  }
  public static int anzahlIterativ(int hoehe) {









  }
}
```

b) Welche Ausgabe hat die Methode `main` der Klasse `Kartenhaus`? (2 Punkte)