Informatik II D-BAUG Prüfung Lösung 4.8.2015 *F. Friedrich*

Name, Vorname: .....

Legi-Nummer: .....

Ich bestätige mit meiner Unterschrift, dass ich diese Prüfung unter regulären Bedingungen ausführen konnte und dass ich die allgemeinen Richtlinien gelesen und verstanden habe.

I confirm with my signature that I was able to take this exam under regular conditions and that I have read and understood the general guidelines.

Unterschrift:

Allgemeine Richtlinien:

General guidelines:

a) Dauer der Prüfung: 120 Minuten.

Exam duration: 120 minutes.

b) Erlaubte Hilfsmittel: Wörterbuch (für gesprochene Sprachen).

Permitted examination aids: dictionary (for spoken languages).

c) Benützen Sie einen Kugelschreiber (blau oder schwarz) und keinen Bleistift. Bitte schreiben Sie leserlich. Nur lesbare Resultate werden bewertet.

Use a pen (black or blue), not a pencil. Please write legibly. We will only correct solutions that we can read.

d) Lösungen sind direkt auf das Aufgabenblatt in die dafür vorgesehenen Boxen zu schreiben. Ungültige Lösungen deutlich durchstreichen. Korrekturen bei Multiple-Choice Aufgaben unmissverständlich anbringen.

All solutions must be written directly onto the exercise sheets in the provided boxes. Invalid solutions need to be crossed out clearly. Corrections to answers of multiple choice questions must be provided without any ambiguity.

e) Falls Sie sich durch irgendjemanden oder irgendetwas gestört fühlen, melden Sie dies sofort der Aufsichtsperson.

If you feel disturbed by anyone or anything, immediately let the supervisor of the exam know this.

f) Wir sammeln die Prüfung zum Schluss ein. Wichtig: stellen Sie unbedingt selbst sicher, dass Ihre Prüfung von einem Assistenten eingezogen wird. Stecken Sie keine Prüfung ein und lassen Sie Ihre Prüfung nicht einfach am Platz liegen. Dasselbe gilt, wenn Sie früher abgeben wollen: bitte melden Sie sich lautlos und wir holen die Prüfung ab. Vorzeitige Abgaben sind nur bis 15 Minuten vor Prüfungsende möglich.

We collect the exams at the end. Important: you must ensure that your exam has been collected by an assistant. Do not take any exam with you and do not leave your exam behind on your desk. The same applies when you want to finish early: please contact us silently and we will collect the exam. Handing in your exam preliminarily is only possible until 15 minutes before the exam ends.

g) Wenn Sie zur Toilette müssen, melden Sie dies einer Aufsichtsperson durch Handzeichen. Es darf zur gleichen Zeit immer nur eine Studentin oder ein Student zur Toilette.

If you need to go to the toilet, raise your hand and wait for a supervisor. Only one student can go to the toilet at a time.

h) Wir beantworten keine inhaltlichen Fragen während der Prüfung. Kommentare zur Aufgabe schreiben Sie bitte auf das Aufgabenblatt.

We will not answer any content-related questions during the exam. Please write comments referring to the tasks on the exam sheets.

Aufgabe	1	2	3	4	5	6	7	8	9	$\sum$
Punkte										
Maximum	10	10	10	10	10	10	10	10	10	90

1 Java *(10 Punkte)*

Betrachten Sie folgenden Code und beantworten Sie die Fragen auf der rechten Seite.

Consider the following code and answer the questions on the right hand side.

```
class A{
    int c = 0;
    static int d = 0; // class variable

    void Update(int value) {
        c += value;
        ++d;
    }

    void Report() {
        System.out.println("c=" + c + ", d=" + d);
    }
}

public class Main {

    static double recursive(double x){
        if (x > 1)
            return ((int)x) * recursive(x-1);
        return x;
    }

    static void first(int a[]){
        int temp = a[0];
        a[0] = a[1];
        a[1] = temp;
    }

    static void second(int a0, int a1) {
        int temp = a0;
        a0 = a1;
        a1 = temp;
    }

    public static void main(String[] args){
        // hier wird (a), (b) oder (c) eingefuegt
        // to be replaced by (a), (b) or (c)
    }
}
```

Im folgenden wird Code angegeben, welcher jeweils in die `main`-Methode der Klasse `Main` auf der linken Seite eingefügt wird. Bitte geben Sie die Ausgabe des Codes direkt in den vorgesehenen Boxen im Code an.

In the following we present code that would be inserted in the `main`-method of the class `Main` on the left hand side. Fill in the output of the code directly into the boxes provided in the code below.

(a) 2 P

```
A a1 = new A(); A a2 = new A();
a1.Update(5); a1.Update(6);
a2.Update(7);
a1.Report(); // output: c = 11 d = 3
a2.Report(); // output: c = 7 d = 3
```

(b) 2 P

```
System.out.println(recursive(2.5)); // output: 1
System.out.println(recursive(5.2)); // output: 24
```

(c) 2 P

```
int a[] = {1,2};
first(a);
int b[] = {1,2};
second(b[0],b[1]);
System.out.println(a[0] + "␣" + a[1]); // output: 2 1
System.out.println(b[0] + "␣" + b[1]); // output: 1 2
```

(d) Die folgende Funktion soll den Ganzzahllogarithmus zur Basis $b > 1$ einer ganzen Zahl $v > 0$ zurückgeben, also das grösste $e \in \mathbb{N}$ so dass $b^e \leq v$. Vervollständigen Sie den Code entsprechend.

The following function is supposed to return the integer logarithm to base $b > 1$ of an integer $v > 0$, i.e. the largest $e \in \mathbb{N}$ such that $b^e \leq v$. Complete the code accordingly. 4 P

```
// pre: positive numbers v (value) and b (base)
// post: return largest integer e such that b^e <= v
static int log(int v, int b){
    int res = 0;
    for (int tmp = b; tmp <= v; tmp *= b)
        ++res;
    return res;
}
```

Ende Aufgabe 1

2 Zufallszahlen (Roulette) *(10 Punkte)*

Betrachten Sie folgenden Code, welcher für die Simulation eines Roulette-Spiels gedacht ist.

Consider the following code intended for the simulation of a roulette game.

```
class Wheel{
    // return a random integer drawn from the numbers {0, ..., 36}
    static int draw(){
        double ran = Math.random(); // returns a random value in [0,1)
        return (int) (ran * 37);
    }
    // return absolute frequencies of a number of draws in an array
    static int[] drawMany(int number){
        int[] frequencies = new int[37];
        for (int i = 0; i<number; ++i)
            ++frequencies[draw()];
        return frequencies;
    }
}
```

---

3 P

(a) Vervollständigen Sie die Methode `Wheel.draw` so, dass sie eine Zufallszahl zurückgibt, welche einer Gleichverteilung über der Menge der ganzen Zahlen $\{0, ..., 36\}$ entstammt.

Complement method `Wheel.draw` such that it returns a random number drawn from a uniform distribution over the set $\{0, ..., 36\}$ of integers.

---

3 P

(b) Vervollständigen Sie die Methode `Wheel.drawMany` so, dass das zurückgegebene Array die absoluten Häufigkeiten von gezogenen Zufallszahlen enthält. Der Parameter `number` enthalte die Anzahl der Zufallszahlen, welche mit der Methode `Wheel.draw` gezogen werden sollen.
Beispiel: Angenommen die vier Aufrufe von `draw()` in `drawMany(4)` geben die Zahlen 1, 3, 5 und 3 zurück, dann sollte `drawMany(4)` folgendes Array zurückliefern: `{0,1,0,2,0,1,0,0,0,...,0}`.

Complement method `Wheel.drawMany` such that the returned array contains the absolute frequencies of drawn random numbers. The parameter `number` contains the number of random numbers that shall be drawn using the method `Wheel.draw`.
Example: *assume that four calls of `draw()` within `drawMany(4)` return values 1, 3, 5 and 3, then `drawMany(4)` should deliver the following array: `{0,1,0,2,0,1,0,0,0,...,0}`.*

(c) Der folgende Code kann verwendet werden, um zu bestimmen ob das einer Zahl zugeordnete Feld rot oder schwarz ist.

The following code can be used in order to determine if a field corresponding to some number is red or black.

4 P

```
class Table{
    boolean[] isRed = new boolean[37];

    Table(){
        for (int i = 0; i < 37; ++i)
            if (i<11 || i>18 && i < 29)
                isRed[i] = (i % 2 == 1);
            else
                isRed[i] = (i % 2 == 0);
    }
    // return if num hits a red field
    boolean Red(int num){
        return isRed[num];
    }
}
```

In folgender Abbildung, welche einen Roulette-Tisch symbolisiert, markieren sie die Felder (ankreuzen), welche gemäss der Methode `Table.Red` rot sind.

In the following figure that stands for a roulette table, mark those fields (with a cross) that are red according to method `Table.Red`.

0	1X	4	7X	10	13	16X	19X	22	25X	28	31	34X
	2	5X	8	11	14X	17	20	23X	26	29	32X	35
	3X	6	9X	12X	15	18X	21X	24	27X	30X	33	36X

Ende Aufgabe 2

3 Objektorientierte Programmierung *(10 Punkte)*

Diese Aufgabe befasst sich mit verschiedenen Aspekten der objektorientierten Programmierung

This task deals with different aspects of object oriented programming.

```
class A [3] {
    String getString(){ return "hello - this is A! "; }
}

class B [4] {
    void printme() {    System.out.println("hello - this is B! "); }
}

class C [4] {
    void printme() {    System.out.print(getString()); }
    String getString(){ return "hello - this is C! "; }
}

class D [6] {
    void printme() {    System.out.println("hello - this is D! "); }
}

public class Main {
    public static void main(String[] args) {
        D[] anArray = new D[4];
        anArray[0] = new A();
        anArray[1] = new B();
        anArray[2] = new C();
        anArray[3] = new D();
        for (int i = 0; i< anArray.length; i ++){
            anArray[i].printme();
        }
    }
}
```

---

4 P (a) Der zuvor gelistete Code soll bei einer Ausführung der main-Methode folgende Ausgabe ergeben:

The code listed before should yield the following output on execution of the main method:

```
hello - this is A! hello - this is B! hello - this is C! hello - this is D!
```

Wählen sie aus den folgenden Textstücken und setzen sie die jeweilige Zahl in den Boxen in obigem Code so ein, dass der Code sich wie gewünscht verhält. (Beachten Sie, dass ein Textstück mehrmals verwendet werden kann. Zahl 6 ist bewusst eine leere und mögliche Option.)

Choose from the following code snippets and provide the according letter in the boxes in the code above such that the code behaves as required. (Note that you can use the same snippet multiple times. Number 6 is intentionally an empty and possible option.)

(1) **extends** A	(4) **extends** D
(2) **extends** B	(5) **extends** Main
(3) **extends** C	(6)

(b) Gegeben sei folgender Code: *Consider the following code:* 3 P

```
//---------------------- File X.java
package foo;
public class X{
    static public void f(){}
    static protected void g(){}
    static void h(){}
    static private void i(){}
}
```

```
//---------------------- File Y.java
package foo;
public class Y {
    void test(){
        X.f();
        X.g();
        X.h();
        X.i();← forbidden
    }}
```

```
//---------------------- File Z.java
package bar;
public class Z extends foo.X{
    void test(){
        f();
        g();
        h();← forbidden
        i();← forbidden
    }}
```

Streichen Sie die Funktionsaufrufe in der Klasse Y und der Klasse Z die aufgrund der Zugriffsmodifizierer in Klasse X nicht erlaubt sind.

Cross out the function calls in class Y and class Z which are not allowed due to the access modifiers in class X

(c) Nennen Sie drei Konzepte der Objektorientierten Programmierung

Name three concepts of object oriented programming 3 P

Kapselung, Vererbung und Polymorphie

Ende Aufgabe 3

4 Verkettete Listen *(10 Punkte)*

Betrachten Sie folgenden Repräsentation einer einfach verketteten Liste.

Consider the following representation of a singly linked list.

```
class ListElement {
    private int data;
    private ListElement next;

    public ListElement(int newData) {
        data = newData;
        next = null;    }
    public void setNext(ListElement nextElem) { next = nextElem; }
    public ListElement getNext() { return next; }
    public int getData() { return data; }
}
```

---

4 P

(a) Vervollständigen Sie die Funktion `mergeSorted` so, dass sie zwei aufsteigend sortierte Listen vereinigt. Resultat ist eine aufsteigend sortierte Liste, welche alle Elementen aus `l1` und `l2` enthält. Vermeiden Sie Kopien der Listenelemente.

Complete the function `mergeSorted` such that it merges two lists, each sorted in ascending order. Result is a list, again sorted in ascending order, containing all elements from `l1` and `l2`. Avoid making copies of list elements.

```
//pre: Lists l1 and l2 sorted in ascending order, l1 or l2 can be empty
//post: merges the two lists l1 and l2 and returns the head of a sorted list
//      containing all elements from l1 and l2.
//      The linked list structure of l1 and l2 is not preserved.
  public ListElement mergeSorted(ListElement l1, ListElement l2){
    ListElement head = new ListElement(0);
    ListElement cur = head;
    while (l1 != null || l2 != null) {
        if (l2 == null || (l1 != null && l1.getData() <= l2.getData())){
            cur.setNext(l1);
            l1 = l1.getNext();
        } else {
            cur.setNext(l2);
            l2 = l2.getNext();
        }
        cur = cur.getNext();
    }
    return head.getNext();
  }}
```

(b) Vervollständigen Sie die Funktion zur Umkehrung einer Liste. Beachten Sie das sie die Funktionen der Liste verwenden sollen. Sie dürfen also keine eigenen Datenstrukuren wie z.B. Arrays hinzunehmen.

Complete the function `reverse`. Note that you are supposed to use the functions of the list to achieve this. Do not use your own data structures, e.g. do not use arrays.

4 P

```
class List {
    public ListElement first, last; //first and last element of the list

     //adds a new list element containing value at the end of the list
    public void append(int value) {...}

     //adds a new list element containing value at the front of the list
    public void insertFirst(int value) {...}

    // pre: singly linked list starting at first
    // post: the same list with the order of elements reversed
    public void reverse() {
        ListElement current = first;
        first = last = null;
        while (current != null) {
            insertFirst(current.getData());
            current = current.getNext();
        }
    }

}
```

(c) Was ist die Komplexität im schlechtesten Fall für das Suchen eines Elementes in einer einfach verketteten sortierten Liste der Länge n?

What is the worst-case complexity for searching an element in a singly linked sorted list with length n?

2 P

$O(n)$ accesses

Ende Aufgabe 4

5 Bäume *(10 Punkte)*

Gegeben sei folgende Repräsentation eines binären Baumes und der nachfolgende Code.

Consider the following representation of a binary tree and the subsequent code.

```
public class Node {
  public int value;
  public TreeNode left;   // all values in left subtree less than value
  public TreeNode right;  // all values in right subtree greater than value

  public TreeNode(int value) {
    this.value = value;
  }
}

public class TreeValidator {

  public static boolean validate(Node node) {
    return validate(root, Integer.MIN_VALUE, Integer.MAX_VALUE);
  }

  protected static boolean validate(Node node, int minValue, int maxValue) {
    if (node == null) {
      return true;
    } else if (node.value <= minValue || node.value >= maxValue) {
      return false;
    } else {
      boolean validLeft  = validate(node.left, minValue, node.value);
      boolean validRight = validate(node.right, node.value, maxValue);
      return validLeft && validRight;
    }
  }

  public static void printPostOrder(Node node) {
    if (node != null) {
      printPostOrder(node.left);
      printPostOrder(node.right);
      System.out.print(node.value + ",");
    }
  }
}
```

(a) Vervollständigen Sie die Methode `validate(...)`, so dass sie `true` zurück gibt wenn es sich bei dem Baum um einen binären Suchbaum handelt. Bei allen anderen Bäumen soll die Funktion `false` zurückgeben. Geben Sie eine rekusive Lösung an.

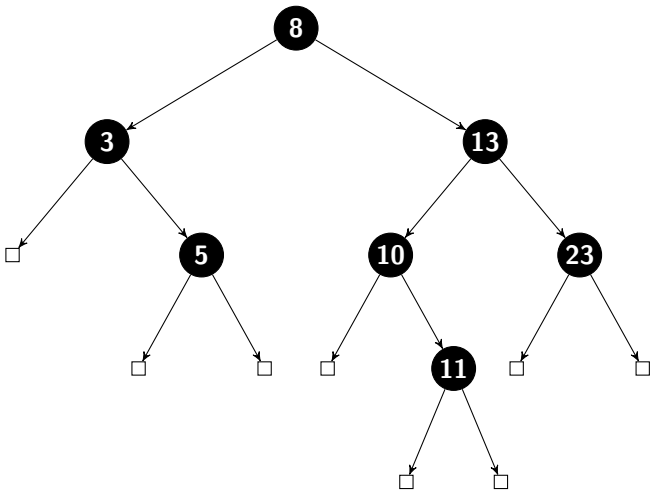
Complete the method `validate(...)` *such that it returns* `true` *if the tree is a valid binary search tree. For all other trees, the function shall return* `false`*. Provide a solution which uses recursion.*

8 P

(b) Die Methode `printPostOrder(Node node)` wird mit der Wurzel des unten dargestellten Baumes aufgerufen. Geben Sie die Ausgabe der Methode an.

The method `printPostOrder(Node node)` *is called on the root node on the tree below. Provide the output of the method.*

2 P




```
5, 3, 11, 10, 23, 13, 8,
```

Ende Aufgabe 5

6 Komplexität *(10 Punkte)*

Betrachten Sie die nachfolgenden Funktionen. *Consider the following functions.*

```
//-------------------------------------------------------------
public double function_1(int n) {
    for (int a = 0; a < n; a++)
        for (int b = 0; b < n; b++)
            functionX(n); //we know the complexity of functionX is O(n)
    return 0.1 + n; }
```

g

```
//-------------------------------------------------------------
public int function_2(int n) {
    int i = n;
    do {
        i -= 2;
    } while (i > 0);
    return n / 2;    }
```

d

```
//-------------------------------------------------------------
public int function_3(int n) {
    int counter = 0;
    for (int a = 0; a < n; a++)
        for (int b = n; b > 0; b--)
            counter++;
    return counter; }
```

f

```
//-------------------------------------------------------------
public String function_4(int n) {
    String output = "";
    for (int a = 1; a <= n; a *= 2) {
        output += a;          }
    return output;   }
```

b

(a) Ordnen Sie den Funktionen auf der linken Seite jeweils die entsprechende präziseste Zeitkomplexität (für die Anzahl Ausführungen elementarer Anweisungen) aus der folgenden Liste zu. Tragen sie den entsprechenden Buchstaben ein.

Assign the functions on the left hand side the corresponding letter of the matching most precise time complexity (for the number of executions of elementary statements) from the following list.

6 P

A) $O(\frac{1}{n})$
B) $O(log(n))$
C) $O(n^{\frac{1}{2}})$
D) $O(n)$
E) $O(n\ log(n))$
F) $O(n^2)$
G) $O(n^3)$
H) $O(2^n)$
I) $O(n!)$

(b) Folgende Funktion `fibonacci` kann benutzt werden, um Fibonacci-Zahlen zu berechnen. Tragen Sie in unten stehende Tabelle ein, wie oft die Funktion jeweils insgesamt aufgerufen wird.

The following function `fibonacci` can be used to compute Fibonacci-numbers. Enter in the table below how often the function will be called eventually.

4 P

```
// pre: n >= 0
// post: return the n-th fibonacci number
public static int fibonacci(int n)
{
    if (n==0)
        return 0;
    else if (n==1)
        return 1;
    else
        return fibonacci(n-1) + fibonacci(n-2);
}
```

Erster Aufruf *first call*	Anzahl Aufrufe *number calls*
`fibonacci(4)`	9
`fibonacci(5)`	15
`fibonacci(6)`	25
`fibonacci(7)`	41

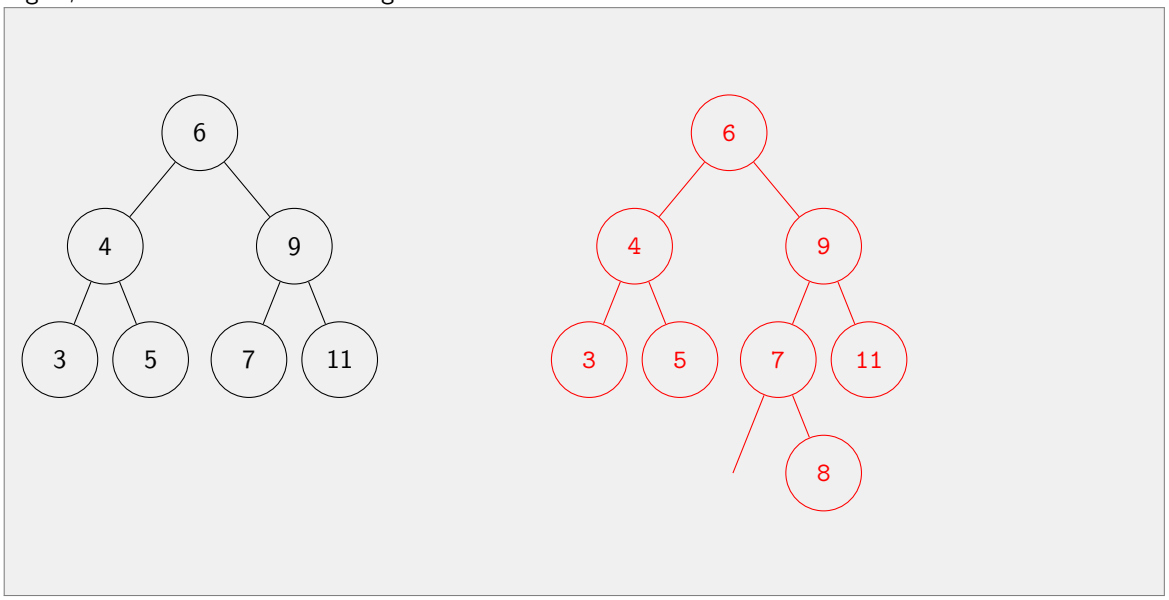
Ende Aufgabe 6

7 Datenstrukturen und Algorithmen *(10 Punkte)*

2 P

(a) Zeichnen Sie neben folgendem **binären Suchbaum** den binären Suchbaum ein, welcher sich nach dem (nicht balancierenden) Algorithmus der Vorlesung ergibt, wenn man die Zahl 8 einfügt.

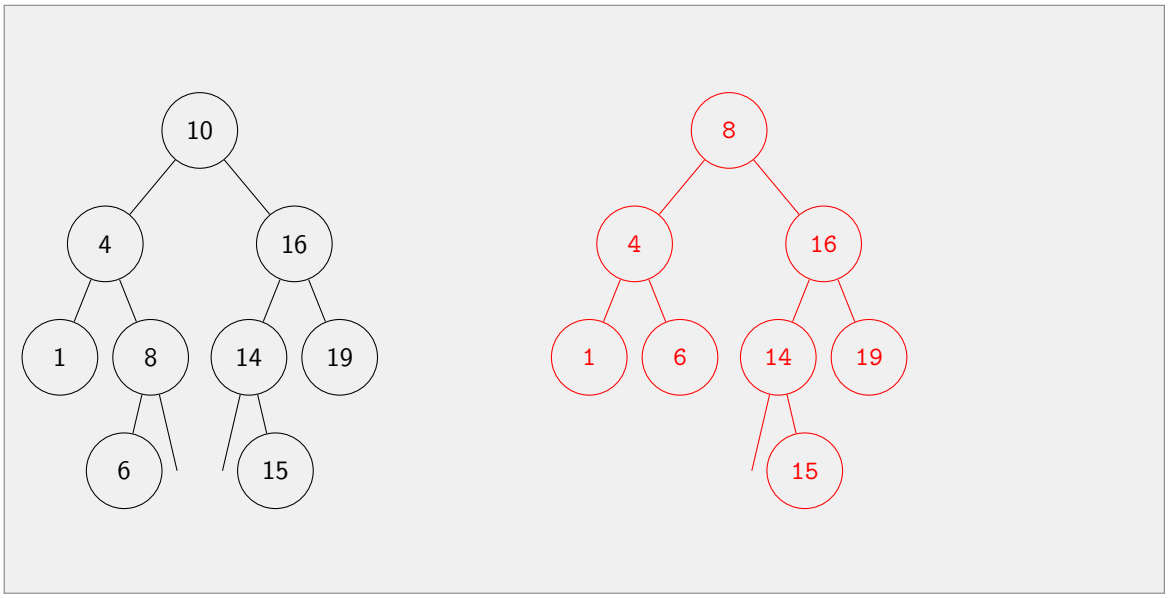
Draw next to the following **binary search tree** *the binary search tree that results from the (non balancing) algorithm shown in the lectures after insertion of the number* 8.



3 P

(b) Zeichnen Sie neben folgendem **binären Suchbaum** einen binären Suchbaum ein, welcher sich nach dem (nicht balancierenden) Algorithmus der Vorlesung ergibt, wenn man die Zahl 10 löscht.

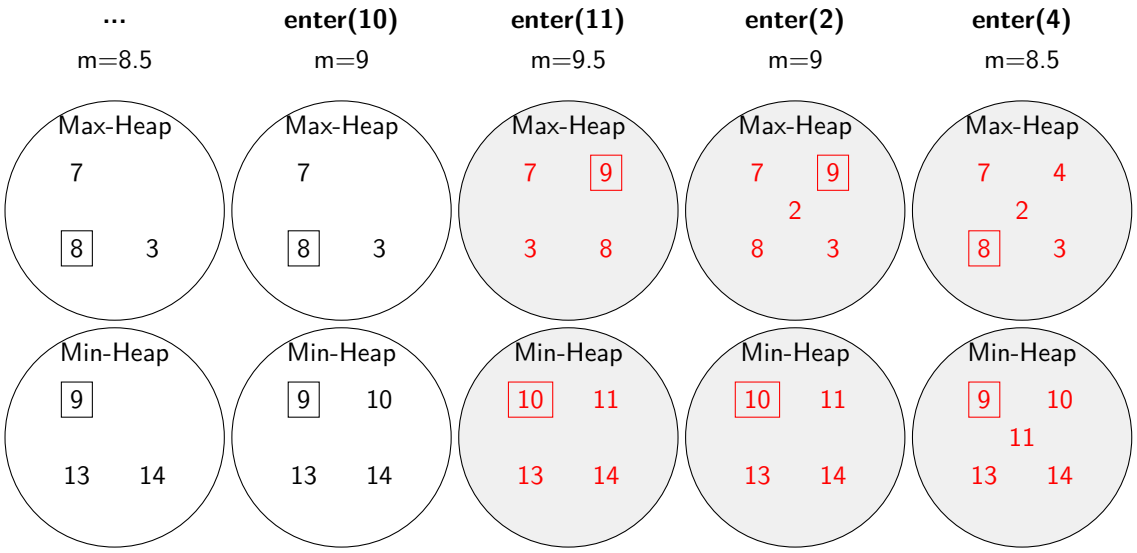
Draw next to the following **binary search tree** *the binary search tree that results from the (non balancing) algorithm shown in the lectures after deletion of the number* 10.



(c) Heaps können für einen Online-Algorithmus des Medians benutzt werden. Im folgenden wird der zugehörige Algorithmus skizziert. Die oberste Zeile beschreibt, welches Element hinzugenommen wird. Die zweite Zeile enthält jeweils den Median. Das jeweils minimale / maximale Element ist durch einen Rahmen hervorgehoben. Bitte vervollständigen Sie die Heaps so wie für die ersten beiden Schritte vorgegeben.

Heaps can be used for an online-algorithm of the median. In the following the corresponding algorithm is sketched. The minimal / maximal element is marked with a frame. The upper most line describes, which element is added to the data structure. The second line contains the median. Please complete the heaps as provided for the first two steps of the algorithm

5 P



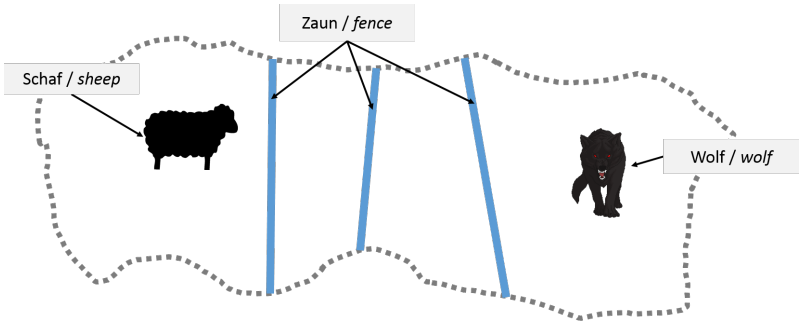
Ende Aufgabe 7

8 Algorithmen *(10 Punkte)*

4 P

(a) Ein Bild enthält folgendes Szenario: eine durch einen Zaun abgetrennte Weide, ein Schaf und ein Wolf. Der Zaun formt ein einfaches Polygon ohne Selbstüberschneidung. Leider ist von dem Bild nur der unten abgebildete Ausschnitt verfügbar. Können Sie trotzdem beurteilen, ob das Schaf sicher ist, also durch den Weidezaun vom Wolf getrennt ist? Kreuzen Sie unten die richtige Antwort an und geben Sie eine Begründung.

A drawing contains the following scenario: a meadow enclosed by a fence, a sheep and a wolf. The fence forms as a simple polygon without self intersection. Unfortunately only the little excerpt of the picture shown below is accessible. Can you nevertheless tell if the sheep is safe, i.e. if it is separated from the wolf by the fence? Mark what is correct and justify your answer below.



x	Das Schaf ist vom Wolf durch den Weidezaun getrennt.	*The sheep is separated from the wolf by the fence.*
	Das Schaf ist nicht vom Wolf durch den Weidezaun getrennt.	*The sheep is not separated from the wolf by the fence.*
x	Es kann nicht entschieden werden, ob das Schaf vom Wolf getrennt steht.	*It is not possible to decide if the sheep is separated from the wolf by the fence.*

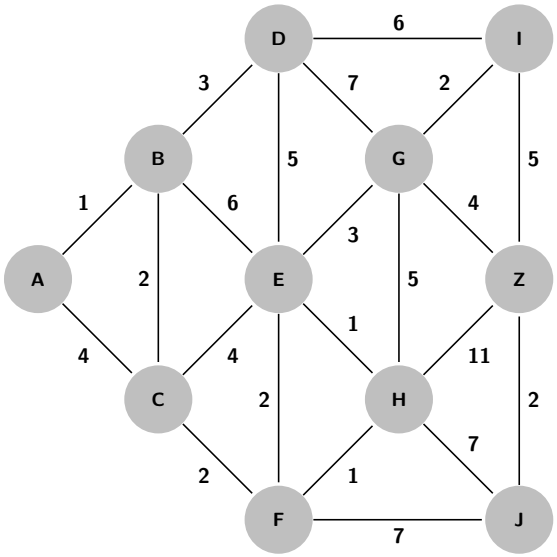
Begründung / *Explanation*

```
Schaf war sicher, Anzahl Überschneidungen entlange einer Geraden ungerade --
PointInPolygonAlgorithmus / JordanKurven.
```

(b) Betrachten Sie folgenden Graphen, welcher alle möglichen Wege zwischen "Städten" A, B, C (etc.) mit deren Längen darstellt. Ziel ist es, einen kürzesten Weg von A nach Z zu finden.

Consider the following graph that shows all possible ways between "cities" A, B, C (etc.) together with their lengths. Goal is to find a shortest path from A to Z.

6 P



Die Zeilen der folgenden Tabelle zeigen die ersten fünf Schritte des Algorithmus von Dijkstra. In jedem Schritt kommt eine Stadt mit bekannter kürzesten Pfadlänge hinzu (hier untersrichen). Zahlen in Klammern bedeuten, dass die jeweilige Stadt in der Nachbarmenge der Städte mit bekanntem kürzesten Pfad liegt. Endliche Zahlen ohne Klammern bezeichnen die definitive kürzeste Weglänge. Vervollständigen Sie die Schritte in der Tabelle.

The rows of the following table represent the first five steps of Dijkstra's algorithm. In each step a city with known shortest path length is added (underlined below). Numbers in brackets mean that the corresponding cities are in the neighboring set of cities with known shortest path. Finite numbers without brackets denote a definitive shortest path length. Complete the steps in the table.

A	B	C	D	E	F	G	H	I	J	Z
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
0	[1]	[4]	∞	∞	∞	∞	∞	∞	∞	∞
0	1	[3]	[4]	[7]	∞	∞	∞	∞	∞	∞
0	1	3	[4]	[7]	[5]	∞	∞	∞	∞	∞
0	1	3	4	[7]	[5]	[11]	∞	[10]	∞	∞
0	1	3	4	[7]	5	[11]	[6]	[10]	[12]	∞

Ende Aufgabe 8

9 Datenbanken *(10 Punkte)*

Im Auftrag der SBB entwickeln Sie ein neues Programm um den Fahrplan zu verwalten. Zu diesem Zweck haben sie folgendes relationale Schema entwickelt:

You are working on a new program to manage the train schedule of the SBB. As part of this project, you created the following database schema:

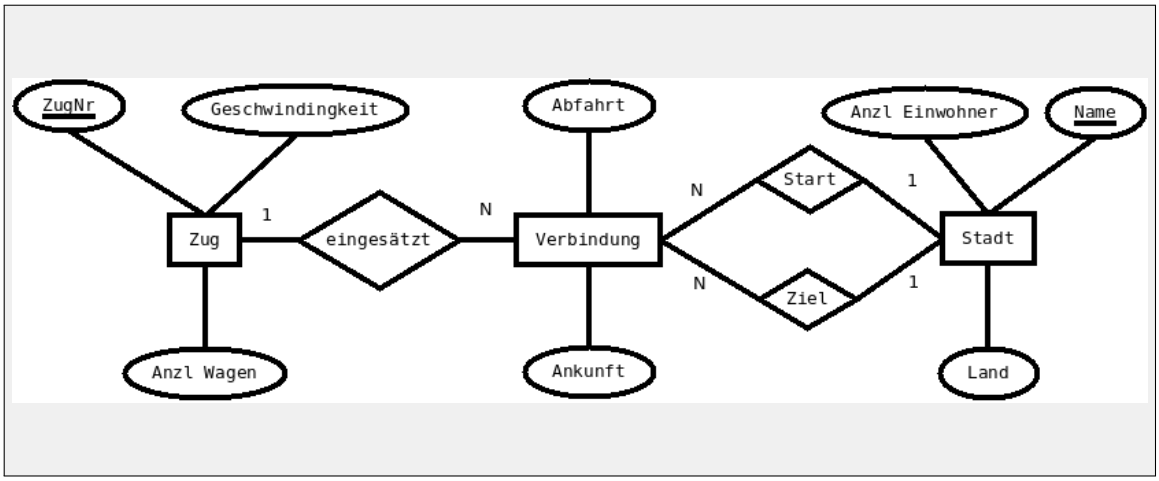
Stadt (City)		
Name	**Einwohner**	**Land**
Zurich	400K	Schweiz
Bern	130K	Schweiz
Genf	200K	Schweiz
Frankfurt	700K	Deutschland
Berlin	3500K	Deutschland
Paris	2200K	Frankreich

Verbindung (Connection)				
ZugNr	**Start**	**Ziel**	**Abfahrt**	**Ankunft**
8001	Zurich	Bern	10:00	11:00
8001	Bern	Genf	11:30	12:30
1540	Frankfurt	Paris	7:00	11:00
9910	Zurich	Berlin	9:00	15:00
9910	Berlin	Zurich	16:00	22:00
4458	Genf	Paris	13:00	16:00

Zug (Train)		
ZugNr	**AnzlWagen**	**Geschwindigkeit**
8001	6	100
1540	8	120
9910	3	180
4458	4	150

---

4 P (a) Geben Sie das entsprechende ER Model an. *Provide the corresponding ER model.*



---

2 P (b) Formulieren Sie folgende Anfrage in SQL:
Was ist die durchschnittliche Geschwindigkeit aller Züge mit mehr als 4 Wagen?

Formulate the following query in SQL:
What is the average speed of all trains with more than 4 coaches?

```
SELECT AVG(Geschwindigkeit) FROM Zug WHERE AnzlWagen > 4
```

(c) Gegeben sind die vier Abfragen (a)-(d). Schreiben Sie die entsprechenden Buchstaben zu den nachfolgenden Antworten. Es gibt Antwortmöglichkeiten ohne entsprechende Abfrage.

Given the four SQL queries (a)-(d), write the corresponding characters next to the results. Some results do not have a mapping query. 4 P

a)
```
SELECT v.Ankunft
FROM Verbindung v
WHERE v.ZugNr = 9910
AND v.Start = 'Berlin'
```

b)
```
SELECT v.ZugNr
FROM Verbindung v, Stadt s1, Stadt s2
WHERE s1.Land <> s2.Land
AND s1.Name = v.Start
AND s2.Name = v.Ziel
ORDER BY v.ZugNr ASC
```

c)
```
SELECT v.Start, Count(*) AS AnzlZuege
FROM Verbindung v
GROUP BY v.Start
HAVING AnzlZuege > 1
```

d)
```
SELECT DISTINCT v.ZugNr
FROM Verbindung v, Stadt s
WHERE s.Land = 'Schweiz'
AND v.Start = s.Name
ORDER BY v.Ankunft ASC
```

	Zurich	*Zurich*
A	22:00	*22:00*
C	Zurich, 2	*Zurich, 2*
	1540, 9910, 4458	*1540, 9910, 4458*
D	8001, 9910, 4458	*8001, 9910, 4458*
	1540, 4458, 9910	*1540, 4458, 9910*
	16:00	*16:00*
	Genf, 1	*Genf, 1*
	9910, 4458, 1540	*9910, 4458, 1540*
B	1540, 4458, 9910, 9910	*1540, 4458, 9910, 9910*
	Paris, Frankfurt, Berlin	*Paris, Frankfurt, Berlin*
	8001	*8001*

Ende Aufgabe 9