

# Informatik II D-BAUG Prüfung Lösung 30.1.2017 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.
- b) Erlaubte Hilfsmittel: Wörterbuch (für gesprochene Sprachen).
- c) Benützen Sie einen Kugelschreiber (blau oder schwarz) und keinen Bleistift. Bitte schreiben Sie leserlich. Nur lesbare Resultate werden bewertet.
- 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.
- e) Falls Sie sich durch irgendjemanden oder irgendetwas gestört fühlen, melden Sie dies sofort der Aufsichtsperson.
- 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.
- 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.
- h) Wir beantworten keine inhaltlichen Fragen während der Prüfung. Kommentare zur Aufgabe schreiben Sie bitte auf das Aufgabenblatt.

- Exam duration: 120 minutes.*
- Permitted examination aids: dictionary (for spoken languages).*
- Use a pen (black or blue), not a pencil. Please write legibly. We will only correct solutions that we can read.*
- 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.*
- If you feel disturbed by anyone or anything, immediately let the supervisor of the exam know this.*
- 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.*
- 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.*
- 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	$\Sigma$
Punkte									
Maximum	10	10	10	10	10	14	10	10	84

# 1 Java (10 Punkte)

Beschreiben Sie in den folgenden Boxen jeweils in Worten, was die darunter stehende Funktion macht.

*Provide in the following boxes what the functions are doing.*

2 P

(a) Ausgabe der Zahlen 1 bis n.

```
// pre: n>=0
static void S(int n){
    if (n>0){
        S(n-1);
        System.out.println(n);
    }
}
```

2 P

(b) Berechnung des Skalarprodukts von a und b.

```
// pre: arrays a and b of same size
static float P(float [] a, float [] b){
    float res = 0;
    for (int i = 0; i<a.length; ++i){
        res += a[i]*b[i];
    }
    return res;
}
```

2 P

(c) Rückgabe des Medians von a,b und c

```
static float M(float a, float b, float c){
    if (a>b)
        if (b>c) return b;
        else if (c>a) return a;
        else return c;
    else
        if (b<=c) return b;
        else if (c<=a) return a;
        else return c;
}
```

- (d) Vervollständigen Sie folgende Funktion so, dass sie die Elemente des Array Parameters a umdreht.

*Complement the following function such that it reverses the elements of the array parameter a.*

3 P

```
static void R(float [] a){
    int u = a.length;
    int l = 0;
```

```
while (--u > l){
    float t=a[u];
    a[u] = a[l];
    a[l++] = t;
}
```

```
}
```

Verwendungsbeispiel:

*Application example:*

```
float [] a = {1,2,3,4};
R(a);
for (float x:a)
    System.out.print(x+" "); // 4.0 3.0 2.0 1.0
```

- (e) Betrachten Sie folgende Klassen- und Variablen-deklarationen.

*Consider the following class and variable declarations.*

1 P

```
class P{
    public int a;
    public P (int A) { a=A; }
}
...
```

```
P p1 = new P(5);
P p2 = new P(5);
P p3 = new P(6);
```

Geben Sie an, ob folgende Ausdrücke wahr (true) oder falsch (false) sind.

*Specify if the following expressions are true or false.*

p1 == p2

false

p2 == p3

false

p1.a == p2.a

true

Ende Aufgabe 1

## 2 Künstliche Wörter (10 Punkte)

Ziel dieser Aufgabe ist die Erstellung künstlicher Wörter aus einem Alphabet. Wir betrachten das Alphabet  $\Omega = \{ ' ', 'A', \dots, 'Z' \}$ . ' ' steht für den leeren Buchstaben.

Aus einem sehr langen Text bestimmt man für jeden Buchstaben  $x \in \Omega$  für alle Folgebuchstaben  $y \in \Omega$  die relative Häufigkeit  $P_{xy}$ . Es ergibt sich eine Wahrscheinlichkeitsmatrix  $(P_{xy})_{x,y \in \Omega}$ .  $P$  enthält also die relative Häufigkeit von Buchstabenkombinationen der Art "AA", "ST", aber auch "\_A" (leerer Buchstabe: Wortanfang) oder "Z\_" (leerer Buchstabe: Wortende).

Nun generiert man aus der Wahrscheinlichkeitsmatrix  $P$  ein Wort, indem man mit dem leeren Buchstaben beginnt,  $x = ' '$ , und mit  $P_x$  so lange neue Buchstaben zufällig zieht, bis wieder ein leerer Buchstabe erscheint.

Nachfolgend sehen Sie einen Ausschnitt aus einer Wahrscheinlichkeitsmatrix, erzeugt mit einem Märchen von Johan Wolfgang von Goethe und einige daraus erzeugte Wörter.

	' '	'A'	'B'	'C'	'D'	'E'	'F'	'G'	'H'	...
' '	0.000	0.073	0.040	0.000	0.154	0.063	0.028	0.047	0.045	...
'A'	0.008	0.002	0.049	0.048	0.006	0.000	0.011	0.047	0.031	...
'B'	0.061	0.061	0.000	0.001	0.000	0.536	0.001	0.015	0.004	...
'C'	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.0902	...
'D'	0.208	0.088	0.000	0.004	0.000	0.431	0.000	0.000	0.000	...
'E'	0.240	0.000	0.014	0.005	0.012	0.001	0.009	0.015	0.022	...
'F'	0.204	0.052	0.000	0.000	0.000	0.167	0.045	0.014	0.039	...
'G'	0.174	0.027	0.000	0.000	0.002	0.514	0.003	0.001	0.000	...
'H'	0.222	0.078	0.001	0.001	0.002	0.165	0.002	0.001	0.003	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

WOPT DEDERUTERM SCHRCHLAGEGOR DIN ASTE FOSITEMER IGTES AUF SENNER DEIMER EMAR ZTENNALENDIN  
 WAHWUR VOCHENS WRINGENE DESTRN ZERERTER ALISCHRDERCH IGSSONSCH VOLIGET ALZUNSCHR LDULE  
 AUNN UN WABGEIE D BR KENUNEIGT SPTCHIHMERR OLIHCHN DE SORBEN UESERER

*Goal of this task is to generate artificial words from an alphabet. We consider the alphabet  $\Omega = \{ ' ', 'A', \dots, 'Z' \}$ . ' ' stands for the empty letter.*

*From a very long text, for each letter  $x \in \Omega$  the relative frequency of all potentially following letters  $y \in \Omega$  is determined as  $P_{xy}$ . This yields a probability matrix  $(P_{xy})_{x,y \in \Omega}$ . Thus,  $P$  contains the relative frequencies for character combinations of the form "AA", "ST", or "\_A" (empty letter before: word starts) or "Z\_" (empty letter after: word ends).*

*Now, from the probability matrix  $P$  a word can be generated by starting with the empty letter,  $x = ' '$ , and subsequently drawing new characters from  $P_x$  until the empty letter appears again.*

*In the following you see an excerpt of a probability matrix  $P$ , generated from a tale from Johann Wolfgang von Goethe and some generated words.*

10 P

- (a) Vervollständigen Sie folgenden Code so, dass der Algorithmus funktioniert.

*Complement the following code such that the algorithm works.*

```

public class Words {
    // 27 characters, first character considered the empty letter
    final static String Letters=" ABCDEFGHIJKLMNOPQRSTUVWXYZ";

    // Sample from the probability vector prob: return an index i from
    // 0 .. prob.length-1 with probability prob[i]
    // pre: prob != null, prob.length > 0
    static int Sample(double prob[]){
        double ran = Math.random(); // returns a random value in [0,1)

        double sum = prob[0];
        int x = 0;
        while (x < prob.length-1 && sum < ran){
            ++x;
            sum += prob[x];
        }
        return x;
    }

    // Generate a new word from the transition (probability) matrix M
    static void GenerateWord(double M[][]){
        int letter = 0;
        do{
            letter = Sample(M[letter]);

            System.out.print(Letters.charAt(letter));
        } while (letter != 0);
        System.out.println();
    }

    // determine the relative frequencies of characters following characters
    // and return corresponding probability matrix
    static double [][] GetMatrix(){
        // implementation omitted for brevity
    }

    public static void main(String[] args) {
        double [][] P = GetMatrix();
        for (int i = 0; i<100; ++i) // generate 100 words
            GenerateWord(P);
    }
}

```

### 3 Dynamisches Array (10 Punkte)

Betrachten Sie folgende Klasse `Vector`, welche ein automatisch adaptierendes eindimensionales Array implementiert.

*Consider the following class `Vector` that implements an automatically adapting dynamic one-dimensional array.*

```
1 public class Vector {
2     int data[] = new int[1]; // invariant: data.length is a power of two
3     int size = 0;           // size of the currently used part of the array
4     // adapt the array size to required number of elements s
5     public void resize(int s){
6         int len = data.length;
7         while (s > len)
8             len *= 2;
9         while (s <= len / 2)
10            len /= 2;
11        if (len != data.length){
12            int tmp[] = new int[len]; // create new object
13            for (int i=0; i<Math.min(len, data.length); ++i)
14                tmp[i] = data[i]; // copy element
15            data = tmp;
16        }
17        size = s;
18    }
19
20    public int get(int ind){
21        if (ind >= size) return 0;
22        return data[ind];
23    }
24
25    public void set(int ind, int value){
26        if (ind >= size){
27            resize(ind+1);
28        }
29        data[ind] = value;
30    }
31
32    public void append(int v){
33        resize(size+1);
34        data[size-1]=v;
35    }
36
37    public void remove(){
38        resize(size-1);
39    }
40    // return the size of the array used
41    public int size() {
42        return size;
43    }
44 }
```

- (a) Geben Sie die Ausgabe zu folgendem Codestück an.

```
Vector v = new Vector();
v.set(2, 10);
v.append(5);
v.set(5, 11);
System.out.print("v=");
for (int i = 0; i < v.size(); ++i)
    System.out.print(" " + v.get(i));
```

*Provide the output of the following piece of code.*

2 P

Ausgabe/output : v= 0 0 10 5 0 11

- (b) Wie oft wird ein neues Array Objekt erzeugt (d.h. wie oft wird Zeile 12 ausgeführt), wenn der folgende Code ausgeführt wird? Und wie viele Elemente des Vektors werden kopiert (d.h. wie oft wird Zeile 14 ausgeführt) when the following code is executed? Tipp: skizzieren Sie sich das Szenario.

```
Vector v = new Vector();
for (int i = 0; i < 16; ++i)
    v.append(i);
```

*How often is a new array object allocated (i.e. how often is line 12 executed) when the following code is run? And how many elements of the vector do have to be copied (i.e. how often is line 14 executed) when the following code is executed? Hint: Sketch the scenario!*

4 P

Allokationen / allocations : 4      Kopien/copies : 15

- (c) Zur Vermeidung von Speicherverschwendung wächst der Vektor nicht nur automatisch, sondern er wird auch automatisch verkleinert, wenn z.B. mittels `remove()` genügend Platz freigegeben wird. Wie oft wird ein neues Array Objekt erzeugt (d.h. wie oft wird Zeile 12 ausgeführt), wenn der folgende Code ausgeführt wird?

```
Vector v = new Vector();
v.set(8, 1);
for (int i = 0; i < 10; ++i){
    v.remove();
    v.append(i*i);
}
```

*In order to avoid a waste of memory the vector is not only automatically increased in size but also automatically shrinks when enough space is freed, e.g. using `remove()`. How often is a new array object allocated (i.e. how often is line 12 executed) when the following code is run?*

2 P

Anzahl Allokationen/number allocations : 21

- (d) Schwierige Frage: Wie würden Sie obigen Code ändern, wenn Sie in Ihrer Anwendung viele `remove` und `append` Operationen erwarten, wenn Kopien und Allokation neuer Objekte und Speicherplatzverschwendung weitgehend vermieden werden soll?

*Tricky Question: How would you change the code above if you expect a lot of `remove` and `append` operations in your application, provided that you want to avoid copies, allocation of new objects and wasting unused memory?*

2 P

Keep binary array sizes and the way how arrays are allocated when increasing the size but only resize when decreasing the size to 1/4 of the previously used size. I.e. replace `len/2` in line 9 by `len/4`.

## 4 Objektorientierte Programmierung (10 Punkte)

Betrachten Sie folgenden Code.

*Consider the following code.*

```
abstract class Base {
    abstract void accept(Visit v);
    abstract String S();
}

class Unary extends Base {
    public int value;
    // constructor
    Unary(int v){ value = v; }
    // return value as string
    String S(){ return "" + value; }
    // call dedicated method of visitor
    void accept(Visit v){ v.visitUnary(this); }
}

class Binary extends Base {
    public Base left;
    public Base right;
    // constructor
    Binary(Base l, Base r){ left = l; right = r; }
    // return string representing this
    String S(){ return "(" + left.S() + "," + right.S() + ")"; }
    // call dedicated method of visitor
    void accept(Visit v){ v.visitBinary(this); }
}

// ----- second part: visitor pattern -----
abstract class Visit {
    abstract void visitUnary(Unary u);
    abstract void visitBinary(Binary b);
}

class E extends Visit{
    public int value = 0;

    void visitUnary(Unary n){ value++; }

    void visitBinary(Binary a){
        a.left.accept(this);
        a.right.accept(this);
    }
}

class F extends Visit{
    public int value = 0;
```



```

void visitUnary(Unary n){ value = n.value; }

void visitBinary(Binary a){
    a.left.accept(this);
    int left = value;
    a.right.accept(this);
    value = left - value;
}
}
}

```

- (a) Geben Sie in folgendem Codeausschnitt jeweils die Ausgaben in den Boxen an. *Provide in the following piece of code the generated output in the boxes.* 6 P

```

Base b = new Unary(10);
System.out.println(b.S());
b = new Binary(new Unary(10), new Unary(20));
System.out.println(b.S());
b = new Binary(b,new Unary(30));
System.out.println(b.S());

E e = new E();
b.accept(e);
System.out.println(e.value);

F f = new F();
b.accept(f);
System.out.println(f.value);

```

10

(10,20)

((10,20),30)

3 [valued 2 points]

-40

- (b) Erlären Sie kurz den Begriff 'Kapselung' / Information hiding: *Explain the term 'encapsulation / information hiding':* 2 P

Verbergen interner Daten und Strukturen = Implementationsdetails vor dem Zugriff von aussen = Interface.

- (c) Neben 'Kapselung', gibt es noch zwei weitere fundamentale Konzepte der objektorientierten Programmierung. Nennen Sie diese. *In addition to 'encapsulation / information hiding', two other fundamental concepts of object-oriented programming exist. Name both.* 2 P

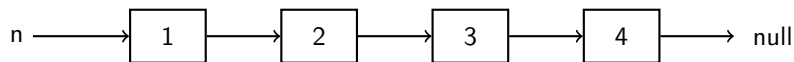
Vererbung, Polymorphie

## 5 Listen und Bäume (10 Punkte)

Es sei eine verkettete Liste von Knoten, die wie folgt definiert sind, gegeben. Das Ende der Liste ist durch einen Null-Knoten gegeben.

```
class Node{
    Node next;
    int value;
}
```

Beispiel:



Assume a linked list of node elements defined as in the following code. The end of the linked list is provided by a null-node.

Example:

Offensichtlich gibt die folgende Funktion die Werte der Knoten, startend beim gegebenen Knoten bis zum Ende der verketteten Liste aus.

Obviously, the following function outputs the node's values starting from the given node element and traversing to the end of the linked list.

```
void Output(Node n){
    while (n != null){
        System.out.println(n.value);
        n = n.next;
    }
}
```

Example Output:

1  
2  
3  
4

4 P

- (a) Vervollständigen Sie den folgenden Code so, dass die Werte in umgekehrter Reihenfolge ausgegeben werden. Tipp: verwenden Sie Rekursion.

Complement the following Code such that the values in the list are output in reversed order. Hint: use recursion.

```
void OutputR(Node n){
    

if (n != null){
            OutputR(n.next);
            System.out.println(n.value);
        }


}
```

Example

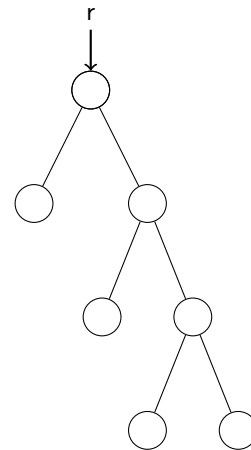
Output:

4  
3  
2  
1

Es sei nun ein binärer Baum gegeben. Die Baumknoten seien wie folgt definiert.

Let now a binary tree be given. Let the tree nodes be defined as follows.

```
public class TNode {  
    int key;  
    TNode left; // linker Teilbaum  
    TNode right; // rechter Teilbaum  
}
```



- (b) Vervollständigen Sie folgende Funktion so, dass sie die Tiefe eines Baumes ab der Wurzel  $r$  zurückgibt.

Complement the following function such that it returns the depth of the tree from root  $r$ .

6 P

```
int GetDepth(TNode r){
```

```
    if (r != null)  
    {  
        int left = GetDepth(r.left);  
        int right = GetDepth(r.right);  
        if (left > right) return left + 1;  
        else return right + 1;  
    }  
    else  
        return 0;
```

```
}
```

## 6 Datenstrukturen und Algorithmen (14 Punkte)

Gegeben seien folgende Daten, eine Sequenz von Zahlen

*Consider the following data, a sequence of numbers.*

1, 3, 4, 2, 6, 5, 7

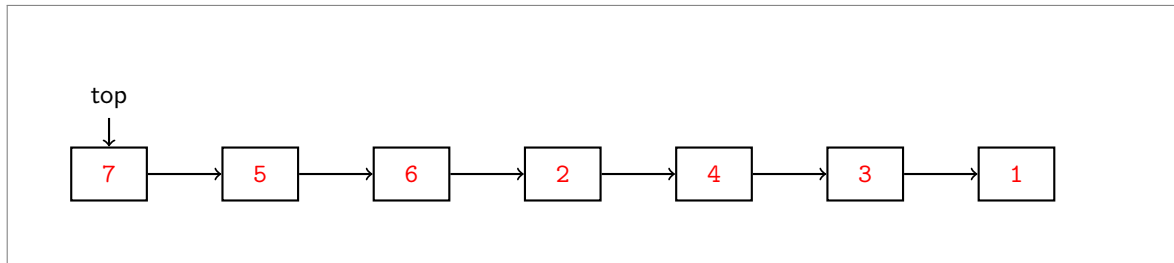
Vervollständigen Sie folgende Figuren, so dass sie deutlich machen, wie die jeweilige Datenstruktur – wie in Vorlesung und Übungen gezeigt – aussieht nach Einfügen der gegebenen Daten in obiger Reihenfolge (von links nach rechts). Vergessen Sie nicht, die Daten (Zahlen) auch einzutragen.

*Complete the following figures such that they illustrate the respective data structures – as presented in lectures and exercises – after insertion of the given data in the order above (from left to right). Do not forget to enter the data (numbers) also.*

1 P

(a) Stack mit verketteter Liste

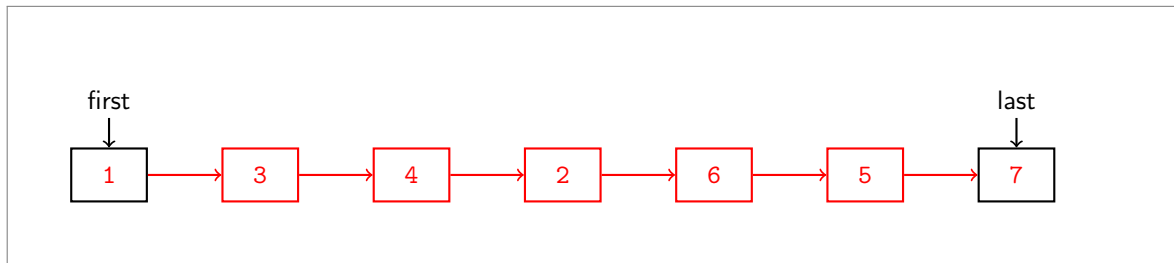
*Stack with linked list*



2 P

(b) Warteschlange (FIFO) mit verketteter Liste

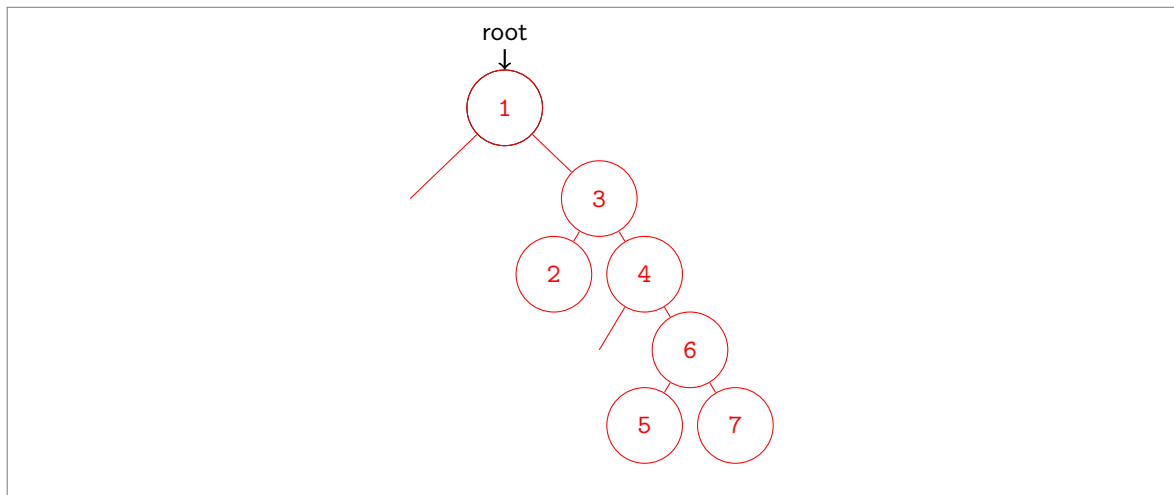
*Queue (FIFO) with linked list*



3 P

(c) Binärer (unbalancierter) Suchbaum

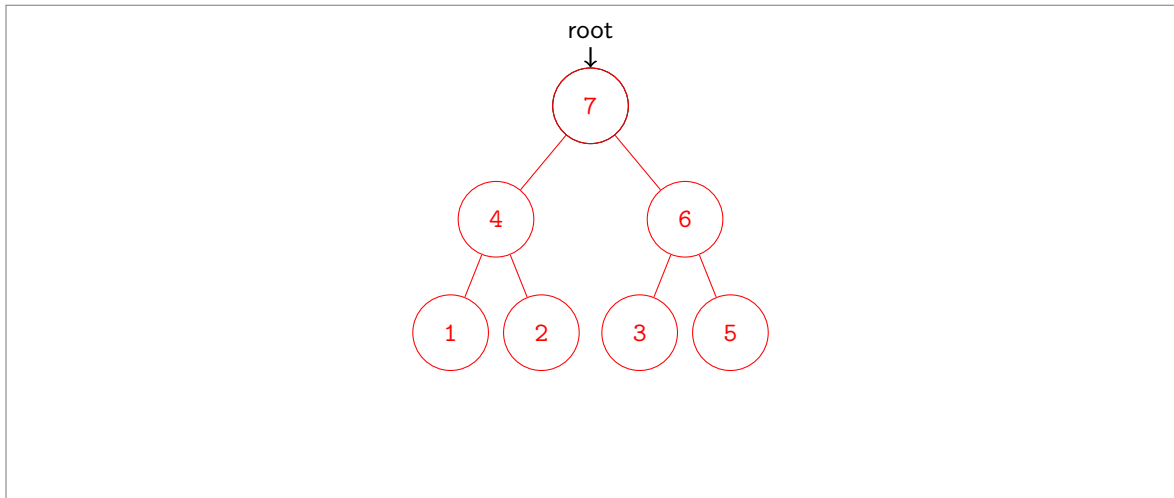
*Binary (unbalanced) search tree*



- (d) Max-Heap (als Baum repräsentiert). Tipp: zeichnen Sie sich die Einzelschritte beim Einfügen auf.

*Max-Heap (represented as Tree). Hint: draw the single insertion steps.*

4 P



- (e) Hashtabelle der fixierten Kapazität 10 unter Verwendung der (schlechten) Hashfunktion  $x \mapsto 2 \cdot x$  und linearer Sondierung. Die oben angegebenen Zahlen bezeichnen jeweils den key, der in der Hashtabelle eingetragen werden soll. Gehen Sie davon aus, dass das Array mit 0 initialisiert wurde und dass 0 für einen leeren Eintrag steht.

*Hashtable with a fixed capacity of 10 using the (bad) hash function  $x \mapsto 2 \cdot x$  and linear probing. The provided numbers represent the key that should be entered in the hash table. Assume that the array is initialized with 0 and 0 stands for the empty entry.*

4 P

table	5	0	1	6	2	7	3	0	4	0
index	0	1	2	3	4	5	6	7	8	9

## 7 Algorithmen (10 Punkte)

1 P

- (a) Sie schreiben ein Programm, welches zu einem immerzu wachsenden Datensatz von reellwertigen Daten jederzeit in hoher Geschwindigkeit den Mittelwert zurückgeben kann. Wie machen Sie das?

*You write a program that to an ever increasing set of real-valued data can provide the mean of the data in very short time. How?*

Provisional means algorithm (basically: keep the value of the mean and the number of data)

2 P

- (b) Beeindruckt von Ihrer Fähigkeit, ein schnelles Programm zur fortwährenden Berechnung des Mittelwertes zu schreiben, werden Sie von Ihrem Boss aufgefordert, dasselbe mit dem Median zu machen. Was schlagen Sie vor?

*Impressed by your ability to implement a program for the continual computation of the mean value, your boss asks you to do the same for the median of the data. What do you suggest?*

Usage of a max- and min-heap with balancing number elements.

1 P

- (c) Ihr Boss bemängelt nun, dass Ihr neuer Algorithmus, obwohl schnell genug, doch speicherintensiv ist. Sie müssen ja alle Daten vorhalten. Lässt sich da etwas machen? Begründen Sie kurz.

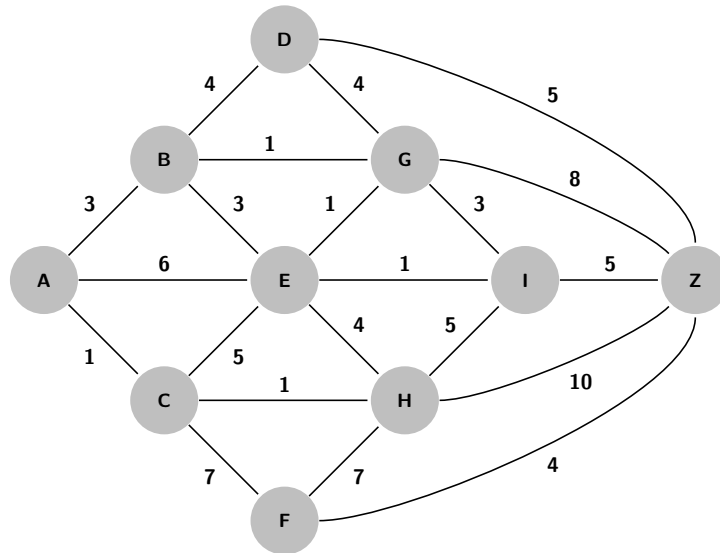
*Now your boss criticizes that your algorithm, although fast enough for its purposes, is memory-intensive. You need to keep all the data. Is there something you can do about it? Give a short explanation.*

No, any value can eventually become a median in the long term. So there is nothing you can do unless you accept a compromise (such as computing the median only of some time window). You might be able to compress the data.

6 P

- (d) 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  $Z$ .*



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

Tipp: Schreiben Sie sich die kürzesten Weglängen auf der Grafik oben mit.

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). 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.

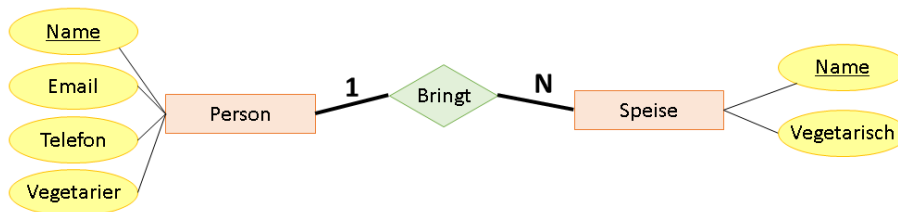
Hint: note down the shortest path lengths in the figure above.

	A	B	C	D	E	F	G	H	I	Z
0	<u>0</u>	[3]	[1]	$\infty$	[6]	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
0	[3]	<u>1</u>	$\infty$	$\infty$	[6]	[8]	$\infty$	[2]	$\infty$	$\infty$
0	[3]	1	$\infty$	$\infty$	[6]	[8]	$\infty$	<u>2</u>	[7]	[12]
0	<u>3</u>	1	[7]	[7]	[6]	[8]	[4]	2	[7]	[12]
0	3	1	[7]	[7]	[5]	[8]	<u>4</u>	2	[7]	[12]
0	3	1	[7]	[7]	<u>5</u>	[8]	4	2	[6]	[12]
0	3	1	[7]	[7]	5	[8]	4	2	<u>6</u>	[11]
0	3	1	<u>7</u>	[7]	5	[8]	4	2	6	[11]
0	3	1	7	[7]	5	<u>8</u>	4	2	6	[11]
0	3	1	7	[7]	5	8	4	2	6	<u>11</u>

## 8 Datenbanken (10 Punkte)

Sie planen eine Party. Sie nehmen die Partygäste in einer Datenbank auf. Dabei speichern Sie zu jeder Person den Namen, Telefonnummer, Email-Adresse, ob sie Vegetarier ist und welche Speise sie mitbringt. Sie achten darauf, dass keine Speise mehrfach mitgebracht wird und halten ob die Speise vegetarisch ist. Sie zeichnen das ER-Diagramm und erhalten folgendes:

*You are planning a party. You store the party guests into a data base. For each person you store name, telephone number, email address, if they are vegetarian and which dish(Speise) they brought. You make sure that no dish will be brought more than once and not if they are vegetarian. You draw the ER-Model and come up with the following:*



2 P

- (a) Geben Sie unten das Relationale Modell zu obigem ER-Diagramm an. Sie können entweder die formale Notation wählen oder Tabellen zeichnen. Fassen Sie ggfs. korrekt zusammen, wie in der Vorlesung gezeigt.

*Provide the relational model to the ER-Diagram above. You can either use a formal notation or draw tables. Combine tables correctly as presented in the lectures.*

```

Personen: {Name, Telefon, Email, Vegetarier}
Speise: {Name, Vegetarisch}
Bringen: {Person, Speise}

Speise und Bringen zusammenfassen ergibt

Personen: {Name, Telefon, Email, Vegetarier}
Speise: {Name, Vegetarisch, mitgebrachtVon}
    
```

2 P

- (b) Formulieren Sie folgende Anfrage in SQL: Wieviel vegetarische und nicht vegetarische Speisen wurden mitgebracht.

*Formulate the following query in SQL: How many vegetarian and non-vegetarian dishes were brought.*

```

SELECT COUNT(*), vegetarisch
FROM Speise
GROUP BY vegetarisch
    
```

2 P

- (c) Formulieren Sie folgende Anfrage in SQL: Welche Personen sind nicht Vegetarier, haben aber eine vegetarische Speise mitgebracht.

*Formulate the following query in SQL: Which persons are non-vegetarians, but brought a vegetarian dish.*

```

SELECT p.name
FROM Person p , Speise s
WHERE s.mitgebrachtVon = p.name
AND p.Vegetarier = FALSE
AND s.vegetarisch = TRUE
    
```



Betrachten Sie folgendes Datenbankschema, welche Kinovorstellungen unterschiedlicher Filme in verschiedenen Kinos enthält.

Consider the following data base scheme that contains movie screenings(Vorstellung) of different movies(Film) in different cinemas(Kino).

<i>Kino (Cinemas)</i>	
KinoID	Name
1	Savoy
2	Rex
3	Corso
4	Abaton

<i>Vorstellung (Screening)</i>		
KinoID	FilmID	Zuschauer
1	1	20
3	1	25
1	4	30
2	5	10
4	2	80
2	3	20
3	2	15

<i>Film (Movie)</i>	
FilmID	Genre
1	Drama
2	Action
3	Action
4	Krimi
5	Drama
6	Krimi
7	Krimi

- (d) Geben Sie zu den folgenden Anfragen jeweils das Resultat an. Tabellen können Sie notieren, indem Sie die Überschriften mit Doppelpunkt, Zeilen mit Semikolon und Spalten mit Komma trennen. Beispiel (Tabelle Kino):  
KinoID,Name: 1,Savoy; 2,Rex; 3,Corso; 3,Abaton.

For each of the following queries provide the result. Tables are denoted by separating the headers with a colon, rows by semicolons and columns by commas. Example (Table Kino):  
KinoID,Name: 1,Savoy; 2,Rex; 3,Corso; 3,Abaton.

4 P

- (A) SELECT k.Name FROM Kino k, Vorstellung v  
WHERE v.Zuschauer > 20 AND k.KinoID = v.KinoID

Name: Corso; Savoy; Abaton

- (B) SELECT f.Genre, COUNT(\*) FROM Vorstellung v, Film f  
WHERE v.FilmID = f.FilmID  
GROUP BY f.Genre

Genre, COUNT(\*): Drama, 3; Action,3 ; Krimi,1

- (C) SELECT SUM(v.Zuschauer) FROM Kino k, Vorstellung v  
WHERE k.KinoID = v.KinoID AND k.Name="Savoy"

SUM(v.Zuschauer): 50

- (D)  $\pi_{Name,Zuschauer,Genre}(Kino \bowtie Vorstellung \bowtie \sigma_{Genre='Drama'}(Film))$

Name, Zuschauer, Genre: Savoy,20,Drama ; Corso,25,Drama ; Rex,10,Drama