

Informatik II D-BAUG Self-Assessment, 30. März 2017 Lösung

Name, Vorname:

Legi-Nummer: Initialien Assistent:

Diese Selbsteinschätzung dient Ihrer und unserer Orientierung. Sie wird eingesammelt, korrigiert und vertraulich behandelt. Sie hat keinen Einfluss auf eine spätere Leistungsbewertung. **Sie haben 20 Minuten Zeit.**

Das folgende Kleingedruckte finden Sie auch auf einer "scharfen" Prüfung.

Allgemeine Richtlinien:

General guidelines:

1. Dauer der Prüfung: 20 Minuten.
2. Erlaubte Unterlagen: Wörterbuch (für gesprochene Sprachen). 4 A4 Seiten handgeschrieben oder ≥ 11 pt Schriftgröße.
3. Benützen Sie einen Kugelschreiber (blau oder schwarz) und keinen Bleistift. Bitte schreiben Sie leserlich. Nur lesbare Resultate werden bewertet.
4. Lösungen sind direkt auf das Aufgabenblatt in die dafür vorgesehenen Boxen zu schreiben (und direkt darunter, falls mehr Platz benötigt wird). Ungültige Lösungen sind deutlich durchzustreichen! Korrekturen bei Multiple-Choice Aufgaben bitte unmissverständlich anbringen! Lösungen auf Notizblättern werden nicht berücksichtigt.
5. Es gibt keine Negativpunkte für falsche Antworten.
6. Störungen durch irgendjemanden oder irgendetwas melden Sie bitte sofort der Aufsichtsperson.
7. 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.
8. Wenn Sie zur Toilette müssen, melden Sie dies einer Aufsichtsperson durch Handzeichen.
9. Wir beantworten keine inhaltlichen Fragen während der Prüfung. Kommentare zur Aufgabe schreiben Sie bitte auf das Aufgabenblatt.

- Exam duration: 20 minutes.*
- Permitted examination aids: dictionary (for spoken languages). 4 A4 pages hand written or ≥ 11 pt font size*
- Use a pen (black or blue), not a pencil. Please write legibly. We will only consider solutions that we can read.*
- Solutions must be written directly onto the exam sheets in the provided boxes (and directly below, if more space is needed). Invalid solutions need to be crossed out clearly. Provide corrections to answers of multiple choice questions without any ambiguity! Solutions on extra sheets will not be considered.*
- There are no negative points for false answers.*
- If you feel disturbed by anyone or anything, let the supervisor of the exam know immediately.*
- 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 ahead of time 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.*
- 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	Σ
Punkte						
Maximum	6	2	6	3	8	25

1 Code Lesen (6 Punkte)

Beschreiben Sie in den folgenden Boxen jeweils in Worten, was die darunter stehende Funktion macht. Beschreiben Sie nicht die einzelnen Schritte sondern die Bedeutung.

Provide in the following boxes what the functions are doing. Do not describe the single steps but rather the meaning.

(a) Ausgabe der Zahlen 1 bis n.

2 P

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

(b) Berechnung des Skalarprodukts von a und b.

2 P

```
// 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;
}
```

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

2 P

```
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; }
}
```

2 Werte und Referenzen (2 Punkte)

- (a) Betrachten Sie folgende Klassen- und Variablendeklarationen.

1 P

Consider the following class and variable declarations.

```
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

3 Asymptotisches Verhalten (6 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

6 P

(a) Ordnen Sie den Funktionen auf der linken Seite jeweils die entsprechende schärfste asymptotische obere Schranke der Anzahl von Elementaroperationen aus der folgenden Liste zu. Tragen sie den entsprechenden Buchstaben von A bis I ein. *Assign to the functions on the left hand side the matching tightest asymptotic upper bound of the number of executed elementary operations from the following list. Enter the corresponding letter from A to I.*

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

4 Rekursion (3 Punkte)

3 P

(a) 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.*

```

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

```

Erster Aufruf <i>first call</i>	Anzahl Aufrufe <i>number calls</i>
fibonacci(4)	9
fibonacci(5)	15
fibonacci(7)	41

5 Die Maus (8 Punkte)

Eine Maus ist jeweils in einem von drei Aufenthaltsorten eingesperrt: Schlafzimmer ($SZ = 0$), Esszimmer ($EZ = 1$), Laufrad ($LR = 2$). Jede Minute werden die Türen kurz geöffnet und die Maus wechselt den Ort oder bleibt. Angenommen, die folgende Wahrscheinlichkeitsmatrix kann das Verhalten der Maus entsprechend dem Markovkettenmodell aus der Vorlesung modellieren.

	SZ	EZ	LR
SZ	0.6	0.3	0.1
EZ	0.3	0.6	0.1
LR	0.1	0.7	0.2

Vervollständigen Sie folgenden Code, so dass er das Verhalten der Maus simuliert und die Gesamtaufenthaltsdauer im Laufrad an einem ganzen Tag ausgibt. Die Maus beginnt im Schlafzimmer (0).

A mouse is locked-in in one of three possible places: sleeping room ($SR = 0$), dining room ($DR = 1$) or in the exercise wheel ($EW = 2$). Once every minute the doors are opened shortly and the mouse can change place or stay. Assume the following probability matrix can model the behaviour of the mouse corresponding to the Markov-Chain model from the lectures:

	SR	DR	EW
SR	0.6	0.3	0.1
DR	0.3	0.6	0.1
EW	0.1	0.7	0.2

Complement the following code such that the behavior of the mouse is simulated. The program should output the duration spent in the exercise wheel of the whole day. The mouse starts in the sleeping room (0).

```

public class Main {

    // sample from the probability vector p:
    // return int k with probability p[k]
    // ( 0 <= k < p.length )
    static int Sample(double[] p){
        double u = Math.random();

        int k=0;
        while (k < p.length && u>0){
            u -= p[k];
            k++;
        }
        return k-1;
    }

    public static void main(String[] args){
        double[][] P = {{0.6,0.3,0.1}, {0.3,0.6,0.1}, {0.1,0.7,0.2}};
        int[] duration = new int[3];
        int current = 0; // Sleeping
        for (int i = 0; i<24*60; ++i){
            duration[current]++;

            current = Sample(P[current]);
        }

        System.out.println("Minutes in hamster wheel: " + duration[2]);
    }
}

```