Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Self-Assessment **(Lösung)**

Informatik II (D-BAUG)

Felix Friedrich

ETH Zürich, April 2019.

Name, Vorname:	. . . . .
Legi-Nummer:	. . . . .

Diese **Selbsteinschätzung** dient Ihrer und unserer Orientierung. Sie wird eingesammelt, korrigiert und vertraulich behandelt. Sie hat aber keinen Einfluss auf eine spätere Leistungsbewertung. **Sie haben 30 Minuten Zeit.**

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

Allgemeine Richtlinien:

1. Dauer der Prüfung: 30 Minuten.
2. Erlaubte Unterlagen: Wörterbuch (für gesprochene Sprachen). 4 A4 Seiten handgeschrieben oder $\geq$ 11pt 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!
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.

General guidelines:

1. *Exam duration: 30 minutes.*
2. *Permitted examination aids: dictionary (for spoken languages). 4 A4 pages hand written or $\geq$ 11pt font size*
3. *Use a pen (black or blue), not a pencil. Please write legibly. We will only consider solutions that we can read.*
4. *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!*
5. *There are no negative points for false answers.*
6. *If you feel disturbed by anyone or anything, let the supervisor of the exam know immediately.*
7. *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.*
8. *If you need to go to the toilet, raise your hand and wait for a supervisor.*
9. *We will not answer any content-related questions during the exam. Please write comments referring to the tasks on the exam sheets.*

Question:	1	2	3	Total
Points:	14	10	6	30
Score:				

Generelle Anmerkung / *General Remark*

Verwenden Sie die Notation, Algorithmen und Datenstrukturen aus der Vorlesung. Falls Sie andere Methoden verwenden, müssen Sie diese kurz so erklären, dass Ihre Ergebnisse nachvollziehbar sind.

Use notation, algorithms and data structures from the course. If you use different methods, you need to explain them such that your results are comprehensible.

Aufgabe 1: Verschiedenes (14P)

1) In dieser Aufgabe sollen nur Ergebnisse angegeben werden. Begründungen sind nicht notwendig.

   In this task only results have to be provided. Explanations are not required.

2) Als Ordnung verwenden wir für Buchstaben die alphabetische Reihenfolge, für Zahlen die aufsteigende Anordnung gemäss ihrer Grösse.

   As the order of characters we take the alphabetical order and for numbers we take the ascending order according to their sizes.

/4P (a) Teilen Sie folgende Algorithmen zu asymptotischen Laufzeiten in Abhängigkeit von der Problemgrösse n zu. Alle Laufzeiten sind für den **schlechtesten Fall**.

Assign the following algorithms to asymptotic running times as a function of the problem size n. All runtimes are for the **worst case**.

- **A** Heap:ExtractMin
- **B** HeapSort
- **C** QuickSort
- **D** MergeSort
- **E** Sortieren durch Auswahl / *Selection Sort*
- **F** Lineare Suche im unsortieren Array / *Linear search in unsorted array*
- **G** Binäre Suche im sortierten Array / *Binary search in sorted array*
- **H** Einfügen in einen AVL Baum / *Insertion into an AVL tree*

$\Theta(1)$	
$\Theta(\log n)$	A, G, H
$\Theta(n)$	F
$\Theta(n \log n)$	B,D
$\Theta(n^2)$	C,E

(b) Im folgenden ist eine Hashtabelle dargestellt (nachdem die Schlüssel 7, 9, 10 und 20 bereits eingefügt wurden). Die Hashfunktion ist $h(k) = k \bmod 13$. Kollisionen werden mit linearem Sondieren aufgelöst. Die Sondierung läuft immer nach rechts. Fügen Sie die folgenden Schlüssel in die (teilweise schon belegte) Hashtabelle in. Wie viele Kollisionen treten beim Einfügen der folgenden drei Zahlen auf?

In the following a hash table is displayed (after keys 7, 9, 10 and 20 had been inserted previously). The hash function is $h(k) = k \bmod 13$. Collisions are resolved using linear probing. Probing always goes left. Insert the following keys into the (partially occupied) hash table. How many collisions take place during the insertion of the three keys?

/4P

Einzufügende Schlüssel/*Keys to be inserted*: 19,6,24

						19	7	20	9	10	**6**	**24**
0	1	2	3	4	5	6	7	8	9	10	11	12

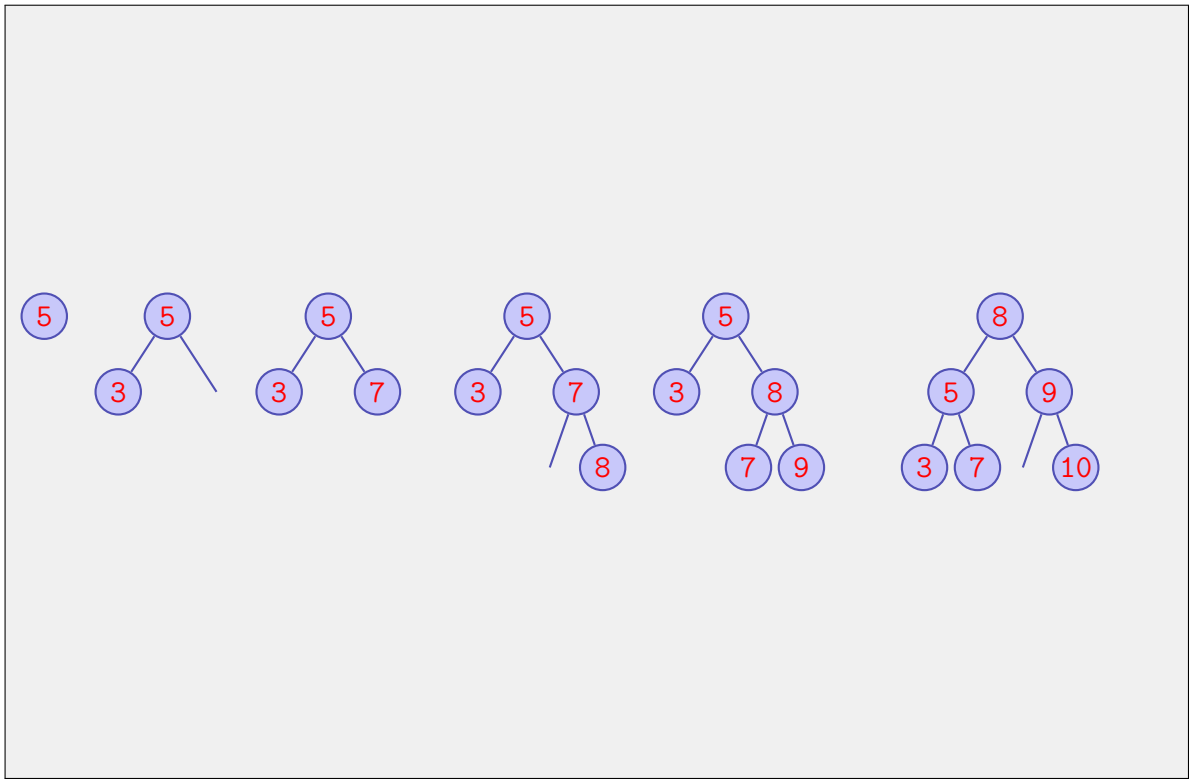
Anzahl Kollisionen/*Number of collisions*: ◯ 4 ◯ 5 √ 6 ◯ 7 ◯ 8

(c) Fügen Sie folgende Zahlen in der angegebenen Reihenfolge in einen initial leeren AVL-Baum ein. Zeichnen Sie den Baum bei jedem Hinzufügen neu.

Insert the following numbers, in the given order, into an initially empty AVL Tree. Redraw the tree for each insertion.

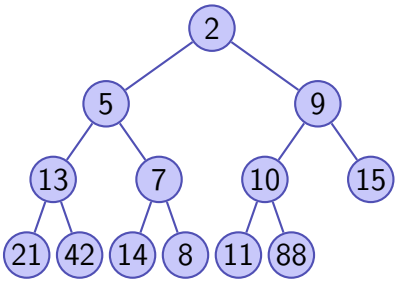
/4P

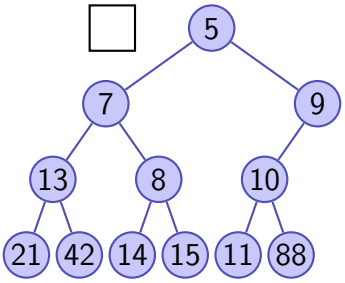
5 3 7 8 9 10

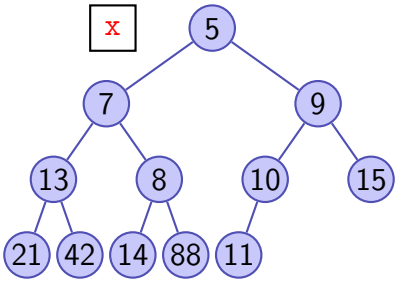


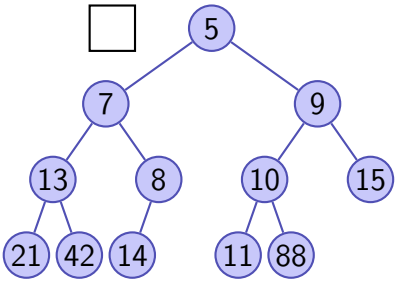
/2P (d) Führen Sie auf folgendem Min-Heap eine Extract-Min Operation aus, wie in der Vorlesung vorgestellt, einschliesslich der Wiederherstellung der Heap-Bedingung. Wie sieht der Heap nach der Operation aus? Kreuzen Sie die richtige Antwort an.

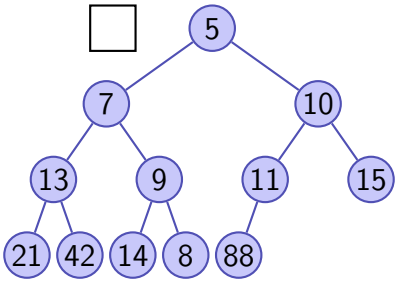
On the following Min-Heap, perform an extract-min operation, including re-establishing the heap-condition, as shown in class. What does the heap look like after the operation? Mark the correct answer.



☐


[x]


☐


☐


Aufgabe 2: Asymptotik (10P)

(a) Finden Sie aus der Liste der in der weissen Box angebenen Funktionen jeweils diejenige, so dass die Gleichheit gilt.

Find from the white boxed list of provided functions for each case below the function such that the equality holds.

/3P

Beispiel/*Example*: $\Theta(n+5) = \Theta(\boxed{n})$

$1,$	$\log n,$	$\log^2 n,$	$n,$	$n \log n,$	$n^2,$	$n^4,$	$n^2 \log n,$	$n \log^2 n,$	$n!,$	n^n

$\Theta(\Sigma_{i=0}^{10n} \log(n^n)) = \Theta(\boxed{n^2 \log n})$

$\Theta(\Sigma_{i=0}^{n^2} i) = \Theta(\boxed{n^4})$

$\Theta(3n + 5n \log n + 0.001n^2) = \Theta(\boxed{n^2})$

In den folgenden Aufgabenteilen wird jeweils angenommen, dass die Funktion g mit $g(n)$ aufgerufen wird. Geben Sie jeweils die asymptotische Anzahl von Aufrufen der Funktion $f()$ in Abhängigkeit von $n \in \mathbb{N}$ mit Θ-Notation möglichst knapp an. Die Funktion f ruft sich nicht selbst auf. Sie müssen Ihre Antworten nicht begründen.

In the following parts of this task we assume that the function g is called as $g(n)$. Provide the asymptotic number of calls of $f()$ depending on $n \in \mathbb{N}$ using Θ notation as tight as possible. The function f does not call itself. You do not have to justify your answers.

/2P (b)

```
void g(int n){
    for (int i = 0; i<100;++i)
        f();
    }
}
```

Anzahl Aufrufe von f / *Number of calls of f*

$\Theta(1)$

/2P (c)

```
void g(int n){
    if (n<100){
        g(n+1);
    }
    f();
}
```

Anzahl Aufrufe von f / *Number of calls of f*

$\Theta(1)$

/3P (d)

```
void g(int n){
    if (n > 1){
        g(n-1)
        g(n-1);
    }
    else{
        f();
    }
}
```

Anzahl Aufrufe von f / *Number of calls of f*

$\Theta(2^n)$

Aufgabe 3: Code Lesen & Schreiben (6P)

/6P

Im Folgenden sehen Sie die Datenstruktur eines Suchbaumknotens.

In the following you see the data structure of binary search tree node.

```
// Data structure to store a Binary Search Tree node
class Node{
    int value;
    Node left = null;
    Node right = null;

    Node(int value) {this.value = value; }
}
```

Ein binärer Baum mit Wurzelknoten `root` wurde aus solchen Knoten zusammengesetzt und Sie wollen wissen, ob er auch ein korrekter Suchbaum ist. Vervollständigen Sie folgenden Code, so dass ein Aufruf an `isBst(root)` zurückgibt, ob der Baum die Suchbaumeigenschaft hat.

A binary tree with root node `root` has previously been componsed of such nodes and you want to know, if it is indeed a correct binary search tree. Complete the follwing code such that a call to `isBst(root)` returns if the tree has the search tree property.

```
// Function to determine if given Binary Tree is a BST or not
public static boolean isBst(Node root){
  return isBstR(root, Integer.MIN_VALUE, Integer.MAX_VALUE);
}
// recursive function called by isBst
public static boolean isBstR(Node node, int min, int max){
  if (node == null) {
     return true;
  }
  if (node.value < min || node.value > max) {
     return false;
  }
  return isBstR( node.left, min, node.value ) &&
         isBstR( node.right, node.value, max );
}
```