

## Probeprüfung (Lösung) Informatik II (D-BAUG)

Felix Friedrich, Hermann Lehner, Departement Informatik

ETH Zürich, Sommer 2020.

Name, Vorname: .....

Legi-Nummer: .....

Ich bestätige mit meiner Unterschrift, dass ich diese Prüfung unter regulären Bedingungen ablegen 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:**

1. Dauer der Prüfung: 90 Minuten.
2. Erlaubte Unterlagen: Wörterbuch (für von Ihnen gesprochene Sprachen). 4 A4 Blätter (beidseitig erlaubt) ohne inhaltliche oder formale Einschränkungen.
3. Benützen Sie einen Kugelschreiber (blau oder schwarz) und keinen Bleistift. Bitte schreiben Sie leserlich. Nur leserliche Resultate werden bewertet.
4. Lösungen sind direkt auf das Aufgabenblatt in den vorgesehenen Boxen zu schreiben (und direkt darunter, falls mehr Platz benötigt wird). Ungültige Lösungen sind nicht durchzustreichen! Korrekturen bei Missverständnissen geben bitte unmissverständlich an.
5. Es gibt keine Negativpunkte für falsche Antworten.
6. Störungen durch irgendjemanden sind nicht erlaubt. Wenn Sie bitte sofort der Aufsichtsperson melden.
7. Wir sammeln die Prüfungen am Ende der Prüfung. Sie legen Sie unbedingt ab und lassen Sie Ihre Unterlagen von den Assistenten eingezogen werden. Dasselbe gilt, wenn Sie vorzeitig abgeben. 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.

1. Exam duration: 90 minutes.
2. Allowed aids: 4 A4 sheets (my be used on both sides) without formal or content-wise restrictions.
3. Use a ballpoint pen (blue or black) and not a pencil. Please write legibly. Only legible solutions that we can read will be evaluated.
4. Solutions should be written directly onto the exam sheet in the provided boxes (and directly below, if more space is needed). Invalid solutions need to be clearly marked. Provide corrections to answers of questions without any ambiguity!
5. There are no negative points for wrong 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	4	5	Total
Points:	19	10	11	8	12	60
Score:						

**Generelle Anmerkung / General Remark**

Verwenden Sie die Notation, Algorithmen und Datenstrukturen aus der Vorlesung. Falls Sie eine andere Herangehensweise wählen, erklären Sie Ihre Antworten in nachvollziehbarer Weise!

*Use notation, algorithms and data structures from the course. If you use a different approach, explain your answers in a comprehensible way!*

**Aufgabe 1: Verschiedenes (19P)**

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.*

- (a) Kreuzen Sie an, ob die folgenden Aussagen wahr oder falsch sind.

*Mark if the following statements are true or false.*

Eine Inorder-Traversierung eines binären Suchbaumes erzeugt eine sortierte Liste der gespeicherten Schlüssel. / *An in-order traversal of a binary search tree generates a sorted list of the stored keys.*

Wahr / True

Falsch / False

Hat eine Folge von  $m$  Operationen im schlimmsten Fall Gesamtkosten  $O(m)$ , dann hat jede einzelne der Operationen im schlimmsten Fall Kosten  $O(1)$ . / *If a sequence of  $m$  operations has overall worst case costs of  $O(m)$ , then every single of the operations has worst cases costs of  $O(1)$*

Wahr / True

Falsch / False

Jeder vergleichbasierte Sortieralgorithmus kann zu einem stabilen Sortieralgorithmus gemacht werden mit asymptotisch, bis auf einen konstanten Faktor, unveränderter Laufzeit. / *Any comparison based sorting algorithm can be made to be stable, without affecting the asymptotic running time by more than a constant factor.*

Wahr / True

Falsch / False

Sei  $G = (V, E)$  ein Graph. Jeder Teilgraph von  $G$  mit  $|V| - 1$  Kanten ist ein Spannbaum von  $G$ . / *Let  $G = (V, E)$  be a graph. Every subgraph of  $G$  with  $|V| - 1$  edges is a spanning tree of  $G$ .*

Wahr / True

Falsch / False

In einem Max-Heap mit  $n$  Schlüsseln ist die Laufzeit zur Extraktion des Minimums im schlechtesten Fall  $O(\log n)$ . / *In a Max-Heap with  $n$  keys the worst-case running time to extract the minimum is  $O(\log n)$ .*

Wahr / True

Falsch / False

In einem AVL-Baum dürfen sich die Anzahlen der Knoten im linken und im rechten Teilbaum maximal um 1 unterscheiden. / *In an AVL-Tree the number of nodes in the left and right subtree must not differ by more than 1.*

Wahr / True

Falsch / False

- (b) Führen Sie auf dem folgenden Array einen Aufteilungsschritt des Sortieralgorithmus Quicksort durch. Benutzen Sie als Pivot das Element 5.

*On the following array, perform a partitioning step of the sorting algorithm Quicksort. As pivot, use the element 5.*

/2P

3	8	10	11	9	5	7	8	2	6	4
0	1	2	3	4	5	6	7	8	9	10
3	4	2	5	9	11	7	8	10	6	8
0	1	2	3	4	5	6	7	8	9	10

- (c) Gegeben sei die folgende Schlüsselmenge:

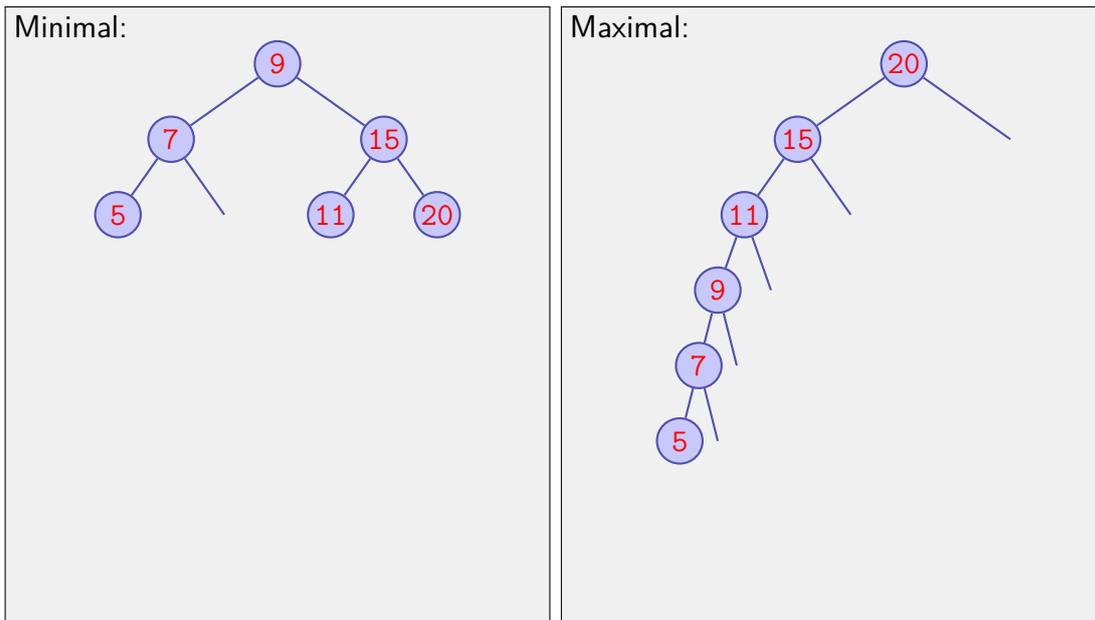
*Let the following set of keys be given:*

/2P

$$K = \{5, 9, 11, 15, 7, 20\}$$

Zeichnen Sie die beiden binären Suchbäume, die genau die Schlüssel aus  $K$  verwalten und die unter allen möglichen Suchbäumen minimale bzw. maximale Höhe haben.

*Draw the two binary search trees that contain the keys from  $K$  and that provide minimal / maximal height.*



- /2P (d) Fügen Sie die folgenden Schlüssel (in der angegebenen Reihenfolge) in die Hashtabelle ein. Verwenden Sie offene Addressierung und lineares Sondieren. Die verwendete Hash-Funktion  $h(k)$  ist nachfolgend angegeben (es wird addiert, also nach rechts sondiert).

*Enter the following keys (in the order provided) into the hash-table. Use open addressing and linear probing. The used hash function  $h(k)$  is provided below (values are added, i.e. probing runs to the right).*

Hashfunktion / *hash function* :  $h(k) = k \bmod 11$

Schlüssel / *keys* : 2,13,24,40

		2	13	24			40			
0	1	2	3	4	5	6	7	8	9	10

Wie viele Kollisionen treten bei der nachfolgenden (erfolglosen) Suche nach dem Schlüssel 100 auf? (Gezählt werden nur Kollisionen mit belegten Plätzen)

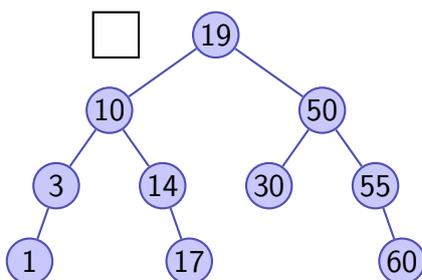
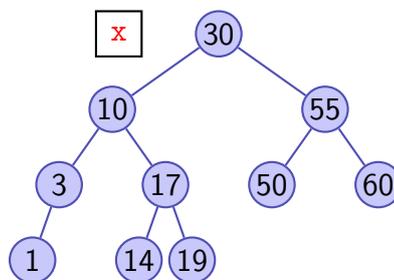
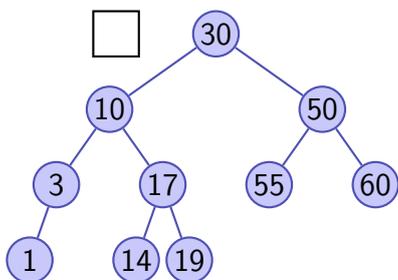
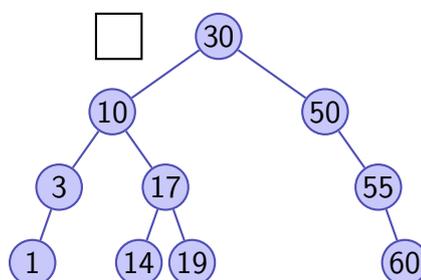
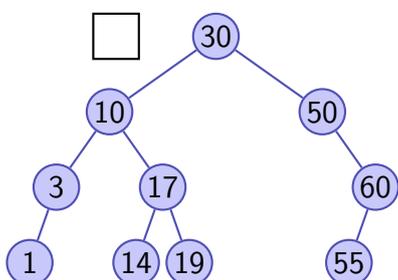
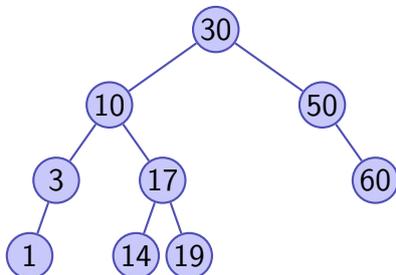
*How many collisions occur when key 100 is searched now (unsuccessfully)? (We only count collisions with occupied spaces)*

0

- (e) Fügen Sie in folgendem AVL Baum den Schlüssel 55 ein und rebalancieren Sie. Wie sieht der AVL Baum nach dem in der Vorlesung gezeigten Algorithmus aus? Kreuzen Sie die richtige Antwort an.

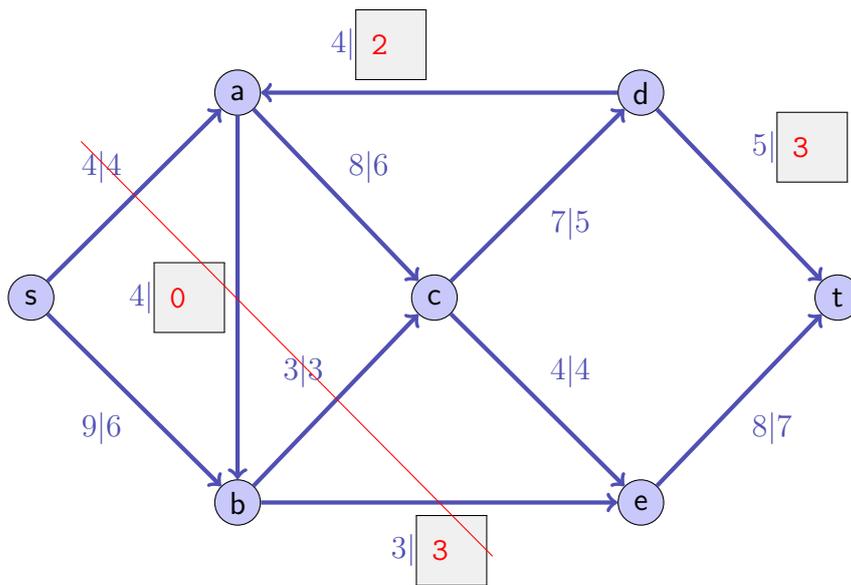
*In the following AVL tree, insert key 55 and rebalance. What does the AVL tree look like according to the algorithms that has been shown in class? Mark the correct answer.*

/2P



/5P (f) Gegeben ist das folgende Flussnetzwerk mit Quelle  $s$  und Senke  $t$ . Die einzelnen Kapazitaten  $c_i$  und Flusse  $\phi_i$  sind an den Kanten angegeben als  $c_i|\phi_i$ . Erganzen Sie die fehlenden Flusswerte auf den Kanten, so dass  $\phi$  ein gultiger Fluss ist. Dieser wird (in diesem Beispiel) in jedem Falle maximal sein. Geben Sie den Wert des Flusses  $f$  an. Zeichnen Sie in der Abbildung einen Schnitt ein, der zeigt, dass  $\phi$  maximal ist.

Provided in the following is a flow network with source  $s$  and sink  $t$ . Capacities  $c_i$  and flows  $\phi_i$  are provided at the edges as  $c_i|\phi_i$ . Complete the missing flow values at the edges such that the overall flow  $\phi$  is a valid flow. This will be maximal (in this example) in all cases. Provide the value of the flow  $f$ . Draw into the figure a cut that shows that  $\phi$  is indeed maximal.



$f = 10$

### Aufgabe 2: Asymptotik (10P)

/3P (a) Geben Sie fur die untenstehenden Funktionen eine Reihenfolge an, so dass folgendes gilt: Wenn eine Funktion  $f$  links von einer Funktion  $g$  steht, dann gilt  $f \in \mathcal{O}(g)$ . Beispiel: die drei Funktionen  $n^3$ ,  $n^5$  und  $n^7$  sind bereits in der richtigen Reihenfolge, da  $n^3 \in \mathcal{O}(n^5)$  und  $n^5 \in \mathcal{O}(n^7)$ .

Provide an order for the functions below such that the following holds: If a function  $f$  is left of a function  $g$  then it holds that  $f \in \mathcal{O}(g)$ . Example: the functions  $n^3$ ,  $n^5$  and  $n^7$  are already in the correct order because  $n^3 \in \mathcal{O}(n^5)$  and  $n^5 \in \mathcal{O}(n^7)$ .

$$\frac{n^2}{\log n}, \quad n \sum_{i=0}^n i, \quad n \sum_{i=0}^{\log n} 2^i, \quad n \log n^n, \quad n\sqrt{n}, \quad n \log n^2,$$

$n \log n^2$	$n\sqrt{n}$	$\frac{n^2}{\log n}$	$n \sum_{i=0}^{\log n} 2^i$	$n \log n^n$	$n \sum_{i=0}^n i$
--------------	-------------	----------------------	-----------------------------	--------------	--------------------

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 Abhangigkeit von  $n \in \mathbb{N}$  mit  $\Theta$ -Notation moglichst knapp an. Die Funktion  $f$  ruft sich nicht selbst auf. Sie mussen Ihre Antworten nicht begrunden.

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

(b)

/2P

```
void g(int n){
    for (double i = 1; i < n; i*=3){
        f();
        i /= 2;
    }
}
```

Anzahl Aufrufe von  $f$  / Number of calls of  $f$  $\Theta(\log n)$ 

(c)

/2P

```
void g(int n){
    if (n > 0){
        g(n-1);
    }
    f();
}
```

Anzahl Aufrufe von  $f$  / Number of calls of  $f$  $\Theta(n)$ 

(d)

/3P

```
void g(int n){
    if (n>0){
        f();
        g(n/2);
        f();
        g(n/2);
    } else {
        f();
    }
}
```

Anzahl Aufrufe von  $f$  / Number of calls of  $f$  $\Theta(n)$

### Aufgabe 3: Algorithmen (11P)

- /1P (a) Um einen kurzesten Pfade Algorithmus auf einem ungewichteten Graphen mit einer asymptotischen Laufzeit zu implementieren, welche linear in Anzahl Knoten und Anzahl Kanten ist, verwendet man im Algorithmus welche Datenstruktur?

- Warteschlange/ *Queue*  
 Stapel/ *Stack*  
 Heap/ *Heap*  
 AVL Baum/ *AVL Tree*

*In order to implement a shortest path algorithm on an unweighted graph with an asymptotic runtime that is linear in number of nodes and number of edges, which data structure is used for the algorithm?*

- /1P (b) Was ist die asymptotische Laufzeit des Bellman-Ford-Algorithmus auf einem vollstandigen Graphen mit  $n$  Knoten?

- $\Theta(n^2)$   
  $\Theta(n \log n)$   
  $\Theta(n^3)$   
  $\Theta(n^4)$

*What is the asymptotic runtime of the Bellman-Ford algorithm on a complete graph with  $n$  nodes?*

- /1P (c) Angenommen wir haben einen korrekt implementierten Kurzeste-Pfade Algorithmus auf einem Graphen mit ganzzahligen Kantengewichten. Wenn wir zu jeder Kante den Wert 1 hinzuaddieren andern sich offensichtlich die Pfadlangen. Aber findet der Algorithmus stets dieselben kurzesten Pfade?

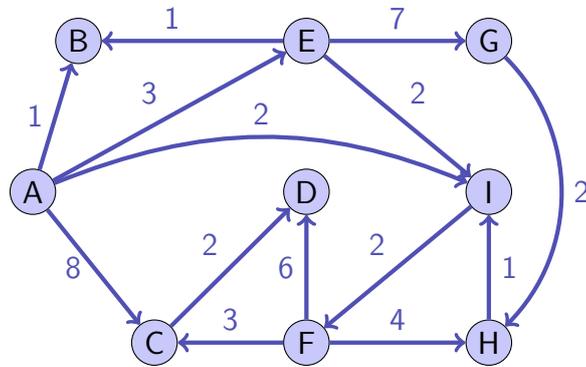
- Ja, bei allen Graphen / *Yes, with all graphs*  
 Ja, bei allen positiv gewichteten Graphen / *Yes, with all positively weighted graphs*  
 Nein / *No*

*Assume you have a correctly implemented shortest-path algorithm on a graph with integer edge weights. If we add 1 to each edge, obviously all path lengths change. But does the algorithm always find the same shortest paths?*

- (d) Betrachten Sie folgenden gerichteten Graph. Geben Sie die Knoten in der Reihenfolge an, in der sie der Dijkstra-Algorithmus besuchen würde, wenn er vom Knoten *A* aus startet. Geben Sie zu jedem Knoten die kürzeste Pfadlänge von *A* aus an.

*Consider the following directed graph. Provide the nodes in the order in which the Dijkstra algorithm visits them when it starts from node *A*. To each node provide the shortest path length from *A*.*

/2P

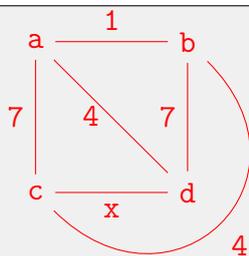


Knoten / <i>Node</i>	Distanz / <i>Distance</i>
A	0
B	1
I	2
E	3
F	4
C	7
H	8
D	9
G	10

- /3P (e) Betrachten Sie folgende Adjazenzmatrix, welche zu einem ungerichteten Graphen mit vier Knoten gehort. Skizzieren Sie zuerst den Graph. Was ist die grosste ganze Zahl fur  $x$ , so dass es ein Paar von Knoten  $a$  und  $b$  gibt, so dass die zu  $x$  zugehorige Kante zwingend auf einem kurzesten Pfad von  $a$  nach  $b$  liegt.

*Consider the following adjacency matrix that corresponds to an undirected graph with four nodes. First sketch the graph. What is the largest integer number for  $x$  such that there is a pair of nodes  $a$  and  $b$  such that a shortest path from  $a$  to  $b$  must include the edge that corresponds to  $x$ ?*

$$\begin{pmatrix} 0 & 1 & 7 & 4 \\ 1 & 0 & 4 & 7 \\ 7 & 4 & 0 & x \\ 4 & 7 & x & 0 \end{pmatrix}$$



Graph

$$x = 8$$

- /3P (f) Sie haben einen sehr grossen Datensatz aus  $n$  unterschiedlichen Zahlen und wollen das  $k$ -kleinste Element finden ( $k \ll n$ ). Wie machen Sie das effizient? Benennen Sie verwendete Datenstrukturen und Laufzeit des Algorithmus.

*You have a huge data set with  $n$  different numbers and you want to find the  $k$ -smallest element ( $k \ll n$ ). How do you do this efficiently? Provide the used data structures and the runtime of the algorithm.*

We use a Max-Heap with  $k$  elements. The first  $k$  elements are filled in. For every following element, we check if it is smaller than the root. If it is smaller than the root, we replace the root by the new element and let it sink (heapify). At the end, the root contains the  $k$ -smallest element.

Runtime  $n \log k$ .

Alternative: use a linked list or array sorted decreasingly with  $k$  elements and insert each new element with a step of bubble sort. Throw away new element. Runtime  $n \cdot k$

Online part: <https://moodle-app6.let.ethz.ch/mod/quiz/view.php?id=2678>

Kennwort / *Password*: Informatik

## Aufgabe 4: Python (8P)

/8P

Diese Aufgabe zum Thema Programmieren in Python soll am Computer gelost werden. Sie konnen sich die Zeit frei einteilen. Wir **empfehlen** aber, dass Sie **nicht mehr als 15 Minuten** fur diese Aufgabe aufwenden.

Losen Sie diese Aufgabe in der Online-Umgebung (Code Expert via Moodle).

*This task on programming with Python needs to be solved at the computer. You are free in how you divide the time. However, we **recommend to spend not more than 15 minutes** on this problem.*

*Solve this task in the online environment (Code Expert via Moodle).*

## Aufgabe 5: Rekursion / Dynamische Programmierung (12P)

/12P

Diese Aufgabe zum Thema Dynamic Programming soll am Computer gelost werden. Sie konnen sich die Zeit frei einteilen. Wir **empfehlen** aber, dass Sie **nicht mehr als 30 Minuten** fur diese Aufgabe aufwenden.

Losen Sie diese Aufgabe in der Online-Umgebung (Code Expert via Moodle).

*This task on programming with Python needs to be solved at the computer. You are free in how you divide the time. However, we **recommend to spend not more than 30 minutes** on this problem.*

*Solve this task in the online environment (Code Expert via Moodle).*

## Anmerkungen zur echten Prufung / *About the real exam*

Bei der echten Prufung stzen Sie sich zu einem Ihnen zugeteilten Rechner und fullen dort ein Formular aus (Name, Legi-Nr, Computernummer, Unterschrift). Sie bekommen die Papierprufung, wahlen am Computer die richtige Prufung aus (der Computer zeigt die Prufungen an, Sie mussen sich keine Webadresse merken) und melden sich am Rechner mit Ihren ETH-Zugangsdaten an. Wir sorgen dafur, dass alle Studierenden zu (fast) derselben Zeit beginnen, indem wir Ihnen allen gemeinsam das Prufungspasswort mitteilen. Die Zeit fur die gesamte Prufung beginnt zu laufen (90 Minuten) und Sie konnen frei einteilen, wann Sie am Papier oder am Computer arbeiten. Nach 90 Minuten werden Sie vom Computer abgemeldet und die Prufung ist auf beiden Seiten (Papier und Computer) zu Ende. / *In a real exam you sit at a computer that has been assigned to you. You fill in a form (name, legi-nr, computer number and signature). You receive the paper exam, choose the right exam (an exam page is displayed at the computer, you do not need to remember any webpage) and log in to the computer with your ETH credentials. We make sure that everybody starts more or less at the same time by providing the exam password to all students simultaneously. Then the individual time for the exam starts to run (90 minutes). You can divide the time freely into work at the*

*computer or work on paper. After 90 minutes you get logged of from the exam and the exam finishes for both the computer an the paper part.*

Mitbringen / *Bring along:*

- Schreibzeug (nicht: Bleistift) / *Writing utensils (not: pencil)*
- ETH Benutzername plus Passwort (im Kopf oder sicher verwahrt) / *ETH username plus password (in your head or stored securely)*

Optional / *Optional*

- Verpflegung / *Food/Drinks*
- Unterlagen (max 4 Blätter) / *examination aids (max 4 sheet)*
- Wörterbuch (falls nötig) / *Dictionary (if required)*