

Tutorium - Informatik 1

Montags 16 st.

Jules Kreuer, VL bei Prof. Grust

📌 Die Lehrevaluation im WS 20/21 läuft! — Bitte gebt uns euer Feedback zur Informatik 1. Danke!

■ Informatik 1 (WS20/21)

🚩 ankündigung

📺 vorlesung




Torsten Grust Professor

1 🗒 1d

Servus zusammen,

wenn ihr in diesen Tagen in eure Mailbox schaut, solltet ihr dort Nachrichten von unserer Studiendekanin finden: **die Lehrevaluation im WS 2020/21** steht an.

Dass dieses Semester speziell ist, brauche ich euch hier nicht vorzutragen:


- noch nie hatten wir so wenig persönlichen Kontakt zu euch in Vorlesungen und Tutorien,
- noch nie hatten wir so wenig Gelegenheit im Hörsaal zu fühlen, wo wir mit der Veranstaltung eigentlich stehen,
- noch nie habe ich stattdessen soviel Zeit auf  verbracht und dann auch noch auf „der anderen Seite der Theke“...

Um so dankbarer sind wir für euer Feedback im Rahmen dieser Vorlesungsevaluation. Bitte nehmt euch ein paar Minuten Zeit und meldet euch zurück. Jedes Bit an Rückmeldung zählt. Die gewonnenen Einsichten sind Gold wert für uns, um einzuschätzen, ob wir die *Informatik 1* in die richtige Richtung steuern und wie wir die Veranstaltung noch besser machen können.

Gerne berichte ich euch in einem der kommenden *Ask Me Anythings* über die Ergebnisse der Evaluation.

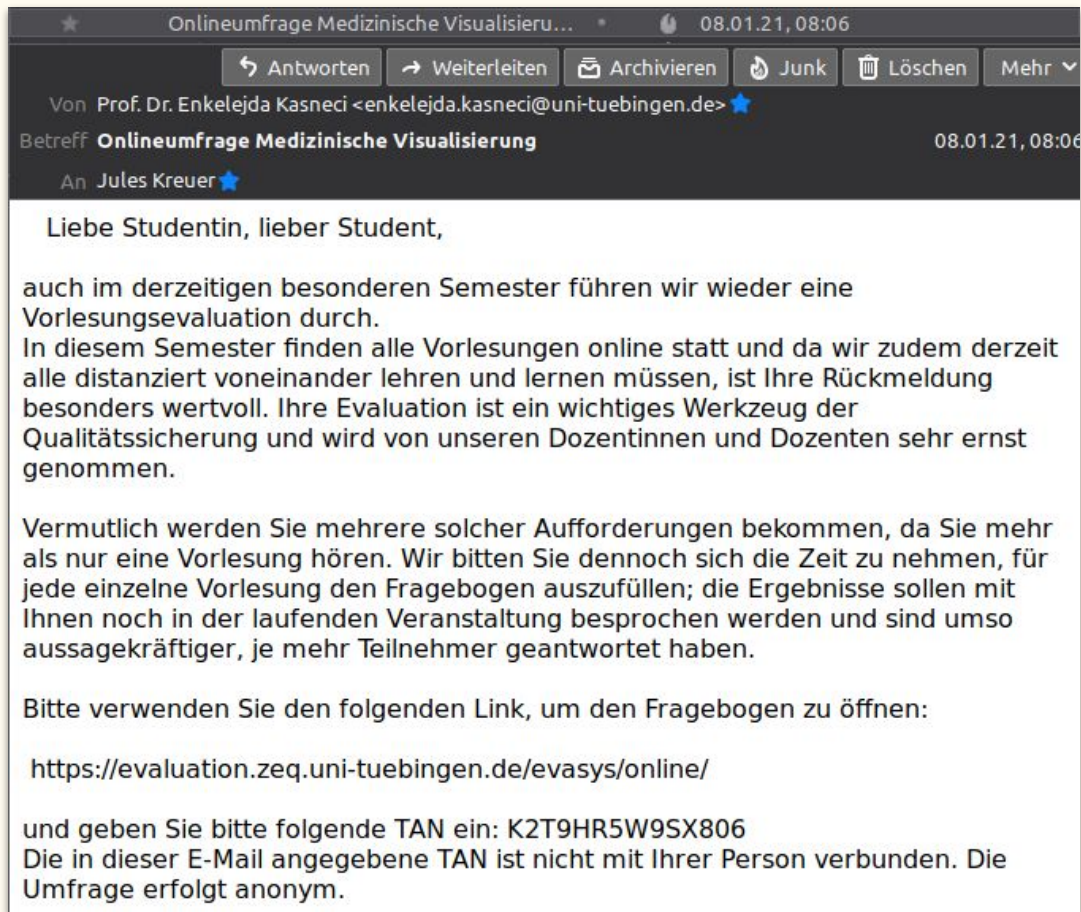
Die Feedbackformulare der Lehrevaluation können

bis zum 15. Januar 2021, 18:00 Uhr

ausgefüllt werden. Nochmals: Von uns schon jetzt ganz herzlichen Dank dafür! 

Beste Grüße,
—TG

<https://forum-db.informatik.uni-tuebingen.de/t/die-lehrevaluation-im-ws-20-21-laeuft-bitte-gebt-uns-euer-feedback-zur-informatik-1-danke/9007>



ÜB 6

Reduktion & Pairs

Prof. Dr. T. Grust, B. Dietrich, C. Duta, D. Hirn

WS 2020/2021

Informatik 1

Forum: <https://forum-db.informatik.uni-tuebingen.de/c/ws2021-info1>

Übungsblatt 6 (16.12.2020)

Abgabe bis: Mittwoch, 23.12.2020, 14:00 Uhr



Relevante Videos: bis einschließlich Informatik 1 - Chapter 07 - Video #031.

<https://tinyurl.com/Informatik1-WS2021>

Sprachebene „Die Macht der Abstraktion — Anfänger“

Aufgabe 1: [10 Punkte]

Gegeben seien die folgenden Definitionen:

```
; Das allumfassende(!) NAND
(: nand (boolean boolean -> boolean))
(define nand
  (lambda (x y)
    (if x (not y) #t)))

; Negation, basierend auf NAND
(: my-not (boolean -> boolean))
(define my-not
  (lambda (x)
    (nand x x)))
```

(a) Reduziere den folgenden Ausdruck:

```
(my-not (< 42 41))
```

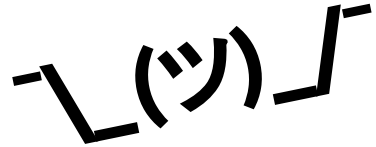
Schwierigkeiten bei ÜB 6

- fast keine :)
- genau lesen*
- rekursions-geschichten → Schauen wir uns genauer an

* zum n -ten mal ;) mit $3 < n \in \mathbb{N}$

Beweise, dass NAND alle logischen Operationen ersetzen kann, indem du Definitionen für die beiden Funktionen

```
(: my-and (boolean boolean -> boolean))  
(: my-or  (boolean boolean -> boolean))
```



angibst, die — ganz analog zu my-not oben — allein auf nand basieren und sich weder auf and oder or beziehen, noch eine Fallunterscheidung (cond, if, match) nutzen.

NAND / NOR

the almighty

Normal Way

a	b	$\neg a$	$\neg b$	\wedge	\vee	\neg
0	0	1	1	0	0	1
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	1	0

NAND Way

$a \neg a$	$b \neg b$	$(a \neg b) \neg (a \neg b)$	$(a \neg a) \neg (b \neg b)$
1	1	0	0
1	0	0	1
0	1	0	1
0	0	1	1

Normal Way

a	b	$\neg a$	$\neg b$	\wedge	\vee	\neg
0	0	1	1	0	0	1
0	1	1	0	0	1	0
1	0	0	1	0	1	0
1	1	0	0	1	1	0

NOR Way

$a \vee a$	$b \vee b$	$(a \vee a) \vee (b \vee b)$	$(a \vee b) \vee (a \vee b)$
1	1	0	0
1	0	0	1
0	1	0	1
0	0	1	1

⇒ NAND und NOR sind allumfassend

Match & Listen

+ rekursion

Match - Allgemein

(match x ← Welches Argument wollen wir matchen
(42 (+ 40 2)) ← Stimmt x mit 42 überein? Wenn ja, gehe in den Zweig
(41 'ney')
((make-test _ a 42) 'record yay'))

record "test"

beliebig

beliebig &
bindung an a

exakt 42

Match - Bindung

```
(match x  
  ((make-test _ a b) (+ (third-val x) 2))  
  ...
```

```
(match x  
  ((make-test _ a b) (+ b 2))  
  ...
```

```
(match x  
  ((make-test __ b) (+ b 2))  
  ...
```

Match - (einfache) Listen

```
(define <name>
  (lambda (.. xs ..)
    (match xs
      (empty ...)
      ...
      ((make-pair y ys) (... (<name> ...))))))
```

i. Schreibt eine Funktion `elem?` mit der Signatur

```
(: elem? (number (list-of number) -> boolean))
```

die eine Zahl und eine Liste von Zahlen erwartet und überprüft, ob die übergebene Zahl ein Element der Liste ist.

Beispiele:

```
(elem? 2 (make-pair 1 (make-pair 2 empty))) ~> #t
```

```
(elem? 3 (make-pair 1 (make-pair 2 empty))) ~> #f
```

iv. Schreibt eine Funktion `reverse` mit der Signatur

```
(: reverse ((list-of %a) -> (list-of %a)))
```

die eine Liste erwartet und deren Elemente in umgekehrter Reihenfolge als Liste zurückgibt.

Hinweis: Verwendet dabei die in Teilaufgabe iii. definierte Funktion `append-elem`.

Beispiel:

```
(reverse (make-pair 1 (make-pair 2 (make-pair 3 empty))))  
  ~> (make-pair 3 (make-pair 2 (make-pair 1 empty)))
```

ÜB 7

Jetzt wirds interessant

Prof. Dr. T. Grust, B. Dietrich, C. Duta, D. Hirn

WS 2020/2021

Informatik 1

Forum: <https://forum-db.informatik.uni-tuebingen.de/c/ws2021-info1>

Übungsblatt 7 (23.12.2020)

Abgabe bis: Mittwoch, 13.01.2021, 14:00 Uhr



Relevante Videos: bis einschließlich Informatik 1 - Chapter 07 - Video #031.

🔗 <https://tinyurl.com/Informatik1-WS2021>

Sprachebene „Die Macht der Abstraktion“

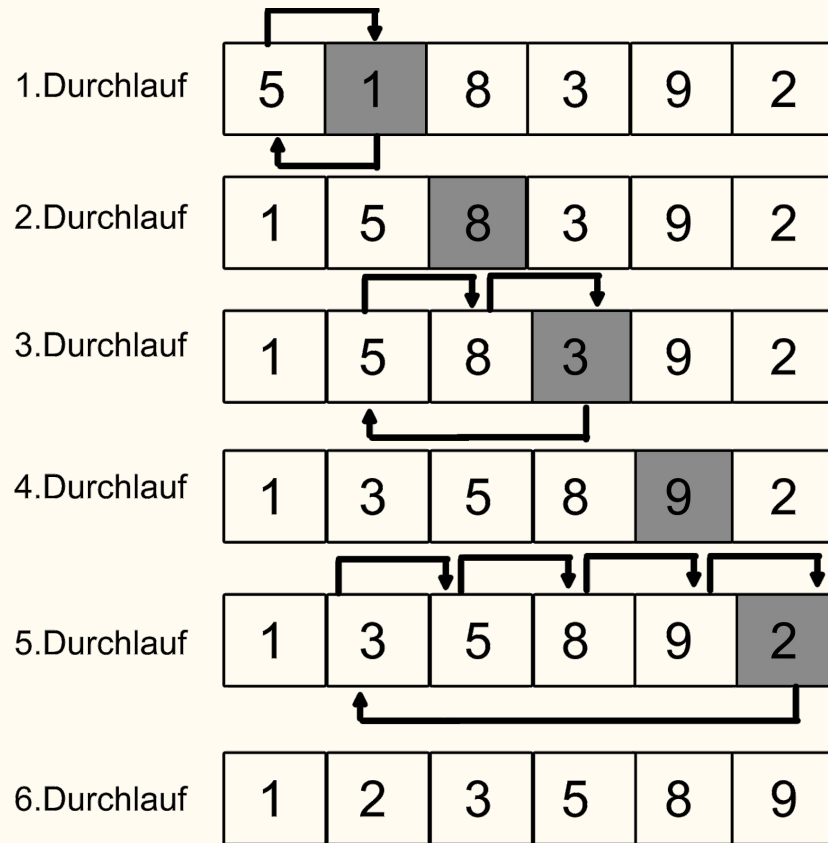
Aufgabe 1: [8 Punkte] Programmiert eine Funktion, die eine Liste von reellen Zahlen akzeptiert und eine Liste ausgibt, die die Zahlen in aufsteigender Reihenfolge sortiert enthält, wobei mehrfach vorkommende Elemente erhalten bleiben.

Geht dazu wie folgt vor:

(a) Schreibt eine Funktion `insert-sorted` mit der Signatur

```
(: insert-sorted (real (list-of real) -> (list-of real)))
```

Die Funktion soll die übergebene Zahl an der richtigen Stelle in die übergebene Liste einfügen und die aufsteigend sortierte Liste zurückgeben. Ihr dürft davon ausgehen, dass die übergebene Liste bereits sortiert ist.



<https://commons.wikimedia.org/wiki/File:InsertionSort.jpg> Mhombach, CC BY-SA 3.0,
via Wikimedia Commons

Feedback:

<https://juleskreuer.eu/feedback>

Feedback

Hier habt ihr die Möglichkeit (anonym) Feedback zu den Tutorien abzugeben. Ich werde versuchen euer Feedback umzusetzen!
Schreibt also ob ich meinen Job gut mache oder ob es Dinge gibt, die man verbessern kann.

i Bleibt bitte konstruktiv!

Fragen? Wünsche? Positives? Negatives?

Vorlesung:

Auswählen ▾

Pseudonym:

Anonym

Absenden