

Algorithmen & Datenstrukturen

Woche 4

Marius Tomek, Nicolas Wehrli, Tim Rieder

17. Oktober 2022

ETH Zürich

Aufgabe 2.4 & Induktion nachbesprechen

Coding conventions

Vorlesung

Abschätzungen

Homework 03 nachbesprechen

Peergrading: Exercise 3.4

Aufgabe 2.4 & Induktion nachbesprechen

Coding conventions

- So divers kann Pseudocode aussehen:
<https://en.wikipedia.org/wiki/Pseudocode#Syntax>
- Kein "richtiger" Code. Soll nicht aussehen wie eine bestimmte Programmiersprache.
- Soll allgemein verständlich sein.
- Soll eindeutig sein.
- Wo könnt ihr nachsehen: Vorlesung, Musterlösungen, Internet.

Pseudocode 2

- Array Index: Oftmals ab 1. Teils aber auch ab 0 beginnend.
- Ihr habt einige Freiheiten.
- Beispiele: `a ← 1` oder `a := 1` oder `a = 1`
- Beispiele: `for i=1..n do` oder `for 1 ≤ i ≤ n`
- Beispiele: `for each elem in array do` oder `for i=1..n do: elem = array[i]`
- Beachtet: Was ist die Laufzeit von `if a in array do` ?

Pseudocode 2

- Array Index: Oftmals ab 1. Teils aber auch ab 0 beginnend.
- Ihr habt einige Freiheiten.
- Beispiele: `a ← 1` oder `a := 1` oder `a = 1`
- Beispiele: `for i=1..n do` oder `for 1 ≤ i ≤ n`
- Beispiele: `for each elem in array do` oder `for i=1..n do: elem = array[i]`
- Beachtet: Was ist die Laufzeit von `if a in array do`? $\mathcal{O}(n)$
- Beachtet: Was ist die Laufzeit von `if a in set do`?

Pseudocode 2

- Array Index: Oftmals ab 1. Teils aber auch ab 0 beginnend.
- Ihr habt einige Freiheiten.
- Beispiele: `a ← 1` oder `a := 1` oder `a = 1`
- Beispiele: `for i=1..n do` oder `for 1 ≤ i ≤ n`
- Beispiele: `for each elem in array do` oder `for i=1..n do: elem = array[i]`
- Beachtet: Was ist die Laufzeit von `if a in array do`? $\mathcal{O}(n)$
- Beachtet: Was ist die Laufzeit von `if a in set do`? Unterschiedlich! Average case bei guter Implementation $\mathcal{O}(1)$ oder $\mathcal{O}(\log n)$
- Beachtet: Was ist die Laufzeit von `for all (a,b,c) in array do`?

Pseudocode 2

- Array Index: Oftmals ab 1. Teils aber auch ab 0 beginnend.
- Ihr habt einige Freiheiten.
- Beispiele: `a ← 1` oder `a := 1` oder `a = 1`
- Beispiele: `for i=1..n do` oder `for 1 ≤ i ≤ n`
- Beispiele: `for each elem in array do` oder `for i=1..n do: elem = array[i]`
- Beachtet: Was ist die Laufzeit von `if a in array do`? $\mathcal{O}(n)$
- Beachtet: Was ist die Laufzeit von `if a in set do`? Unterschiedlich! Average case bei guter Implementation $\mathcal{O}(1)$ oder $\mathcal{O}(\log n)$
- Beachtet: Was ist die Laufzeit von `for all (a,b,c) in array do`? $\mathcal{O}(n^3)$

IMMER bei Coding-Aufgaben:

- Ihr müsst Laufzeit und Korrektheit beweisen oder zumindest argumentieren/begründen.

WENN es explizit steht:

- Ihr müsst Laufzeit oder Korrektheit auch formal beweisen. Das seht ihr dann (Invarianten).

Vorlesung

- Linear Search. Laufzeit?

- Linear Search. Laufzeit? $\mathcal{O}(n)$
- Binary Search. Laufzeit?

- Linear Search. Laufzeit? $\mathcal{O}(n)$
- Binary Search. Laufzeit? $\mathcal{O}(\log n)$
- Is Sorted. Laufzeit?

- Linear Search. Laufzeit? $\mathcal{O}(n)$
- Binary Search. Laufzeit? $\mathcal{O}(\log n)$
- Is Sorted. Laufzeit? $\mathcal{O}(n)$
- Bubblesort:

- Linear Search. Laufzeit? $\mathcal{O}(n)$
- Binary Search. Laufzeit? $\mathcal{O}(\log n)$
- Is Sorted. Laufzeit? $\mathcal{O}(n)$
- Bubblesort: Kleinstes Element nach hinten rotieren.
- Selection Sort:

- Linear Search. Laufzeit? $\mathcal{O}(n)$
- Binary Search. Laufzeit? $\mathcal{O}(\log n)$
- Is Sorted. Laufzeit? $\mathcal{O}(n)$
- Bubblesort: Kleinstes Element nach hinten rotieren.
- Selection Sort: Nächstes richtiges Element selektieren.
- Insertion Sort:

- Linear Search. Laufzeit? $\mathcal{O}(n)$
- Binary Search. Laufzeit? $\mathcal{O}(\log n)$
- Is Sorted. Laufzeit? $\mathcal{O}(n)$
- Bubblesort: Kleinstes Element nach hinten rotieren.
- Selection Sort: Nächstes richtiges Element selektieren.
- Insertion Sort: Nächstbestes Element am richtigen Ort einsortieren.

Invariante (einführen) am Beispiel Bubble-/Selection-/Insertion Sort.

Abschätzungen

Aus alter Prüfung: <https://exams.vis.ethz.ch/exams/z9ojf5sb.pdf>

Homework 03 nachbesprechen

Peergrading: Exercise 3.4
