

Algorithmen und Datenstrukturen

A1. Organisatorisches

Marcel Lüthi and Gabriele Röger

Universität Basel

3. März 2021



Algorithmen und Datenstrukturen

3. März 2021 — A1. Organisatorisches

A1.1 Organisatorisches

A1.2 Über diese Vorlesung

A1.1 Organisatorisches

Personen: Dozenten



Marcel Lüthi



Gabriele Röger

Dozenten

Dr. Marcel Lüthi

- ▶ **E-Mail:** marcel.luethi@unibas.ch
- ▶ **Büro:** Raum 04.002, Spiegelgasse 1

Dr. Gabriele Röger

- ▶ **E-Mail:** gabriele.roeger@unibas.ch
- ▶ **Büro:** Raum 04.005, Spiegelgasse 1

FFFFF 6bfa01a2b87efc8b48ba167c2a031d7a768e354b

Wahl der Übungsgruppe

- ▶ Bitte tragen Sie Ihre Präferenzen in <https://courses.dmi.unibas.ch> ein.
- ▶ Es müssen alle 3 Präferenzen angegeben werden.

Vorlesung im Web

Vorlesungsseite

<https://dmi.unibas.ch/de/studium/computer-science-informatik/lehrangebot-fs21/vorlesung-algorithmen-und-datenstrukturen/>

- ▶ Vorlesungsbeschreibung
- ▶ Folien
- ▶ Zusatzmaterial (nicht prüfungsrelevant)

Anmeldung:

- ▶ <https://services.unibas.ch/>
- ▶ Bitte registrieren Sie sich gleich heute, um alle kursrelevanten Informationen zu erhalten.
- ▶ Bitte tragen Sie sich auch für eine Übungsgruppe ein (unter <https://courses.dmi.unibas.ch>). → jetzt

Vorlesungsmaterialien

Vorlesungsmaterialien:

- ▶ Vorlesungsfolien (online)
- ▶ Lehrbuch
- ▶ vertiefendes Material **auf Anfrage**

Lehrbuch



Algorithmen
von Robert Sedgewick und Kevin Wayne
(Pearson Verlag, 4. Auflage)

Weitere nützliche Ressourcen

- ▶ Seite zum Buch: Algorithms, 4th edition:
<https://algs4.cs.princeton.edu/home/>
- ▶ Youtube (Suchbegriff: Robert Sedgewick algorithms)
- ▶ Data Structures and Algorithms – The Basic Toolbox von Kurt Mehlhorn und Peter Sanders (Springer Verlag)
<http://people.mpi-inf.mpg.de/~mehlhorn/Toolbox.html>
- ▶ Google, Wikipedia, ...

Zielgruppe

Zielgruppe:

- ▶ Bachelor Informatik (ab 2. Semester)
- ▶ Bachelor Computational Sciences (ab 2. Semester)
- ▶ Alle Studierenden mit Programmierkenntnissen sind herzlich willkommen.

Voraussetzungen:

- ▶ Programmierung (Java)

Programmiersprachen

- ▶ Vorlesung: Hauptsächlich Python
 → Vorteil: Kompakt und direkt, ideal für kleine Programme
- ▶ Übungen: Java oder Python (nach Ankündigung)



Es werden keine Python-Kenntnisse vorausgesetzt!

Übungen

Übungsaufgaben:

- ▶ Hausaufgaben (Theorie + Praxis)

Übungstermine:

- ▶ Vorbesprechung der Hausaufgaben
- ▶ Beantwortung von Fragen zum aktuellen Blatt
- ▶ Technische Hilfestellung (Java/Python, Programmierumgebung)
- ▶ Nachbesprechung der Übungen
- ▶ Teilnahme freiwillig - **aber sehr empfohlen**.

Übungen: Hausaufgaben

Hausaufgaben:

- ▶ Aufgaben ab Donnerstagabend auf Adam verfügbar.
- ▶ Bearbeitung in Zweiergruppen ($2 \neq 3$)
- ▶ Abgabe freitags in Folgewoche (23:59) auf Adam
- ▶ Besprechung und [individuelles Feedback](#) in Übungsgruppe

Discord-Server

- ▶ Fragen können auf dem Vorlesungs-Discord-Server diskutiert werden.
 - ▶ Idee: Kursteilnehmer helfen sich gegenseitig.
 - ▶ Dozierende können auf Anfrage mithelfen.

Erklären und helfen heisst lernen!

Prüfung

- ▶ **schriftliche Prüfung**
- ▶ voraussichtlich Mo, 14. Juni 2020, 14-16 Uhr
- ▶ 8 Kreditpunkte
- ▶ Zulassung:
 - ▶ Alle bis auf höchstens drei Übungsblätter (8 von 11) erfolgreich bearbeitet
 - ▶ „erfolgreich bearbeitet“ = mind. 60% der Punkte
- ▶ Note basiert nur auf Klausur
- ▶ **keine Wiederholungsprüfung**

Prüfungsvorbesprechung am letzten Vorlesungstermin
(3.6.2021, 14:15-16:00)

Plagiate

Plagiat (Wikipedia)

Ein Plagiat ([...]) aus lateinisch *plagiarius*, deutsch „Seelenveräufer, Menschenräuber“) ist die Anmaßung fremder geistiger Leistungen. Dies kann sich auf die Übernahme fremder Texte oder anderer Darstellungen [...], fremder Ideen [...] oder beides gleichzeitig [...] beziehen.

Folge:

- ▶ 0 Punkte für Übungsblatt (beim ersten Mal)
- ▶ Nicht-Zulassung zur Prüfung (im Wiederholungsfall)

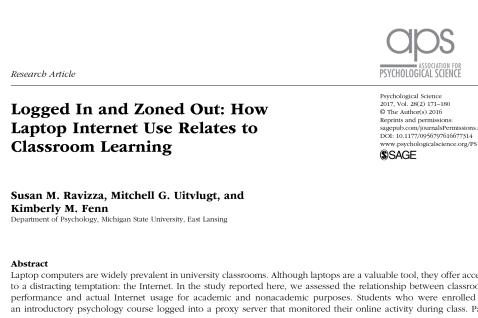
[Hilfe von Kolleg*innen/Internet?](#) Gerne. Aber Lösungscode nicht anschauen und **niemals kopieren**.

[Aufgaben zu schwer?](#) Wir helfen gerne!

Laptops

Wir sitzen offensichtlich vor einem Computer.

Aber Vorsicht:

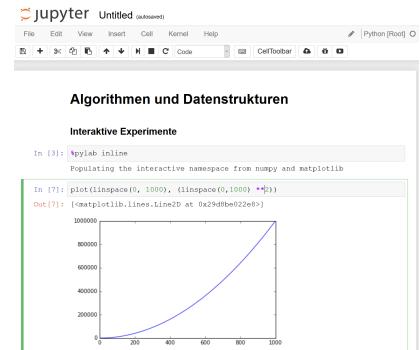


Jupyter-Notebooks

Web-basierte interaktive Arbeitsumgebung für Python

Nutzung von Jupyter-Notebooks:

- ▶ Erklärungen zu Algorithmen
- ▶ Implementation von Algorithmen
 - ▶ Ideal zum Experimentieren und Lernen
- ▶ Miniübungen während der Vorlesung



Jupyter-Notebooks – Installation

- ▶ (Einfache) manuelle Installation erforderlich
- ▶ Empfehlung: Installation via Anaconda
 - ▶ Gleiche Umgebung wie bei Erweiterte Grundlagen der Programmierung
- ▶ Installationsanleitung:
<https://jupyter.readthedocs.io/en/latest/install/notebook-classic.html>

Fragen zur Organisation

Fragen?

A1.2 Über diese Vorlesung

Algorithmen und Datenstrukturen

- ▶ Bestimmte Grundbausteine benötigt man immer wieder bei Programmierprojekten, z.B.
 - ▶ Sortierverfahren
 - ▶ Suchbäume
 - ▶ Prioritätswarteschlangen
 - ▶ kürzeste Pfade in Graphen
 - ▶ ...
- ▶ Wird oftmals durch Bibliotheken fertig bereitgestellt.
- ▶ Hier lernen Sie ...
 - ▶ wie das alles intern funktioniert.
 - ▶ wie man den richtigen Baustein auswählt.
 - ▶ Tricks und Kniffe für effiziente Programme.
- ▶ Methoden unabhängig von konkreter Programmiersprache

Beispiel: Sortieralgorithmen

- ▶ Aufgabe: Bringe Sequenz von Elementen in aufsteigende Reihenfolge, z.B.
Eingabe $[5, 9, 3, 5] \rightarrow$ Ausgabe $[3, 5, 5, 9]$
- ▶ 1960er Jahre (und noch lange danach): ein Viertel der kommerziell verbrauchten Rechenzeit für Sortievorgänge
- ▶ Naiver Algorithmus: **Selectionsort**

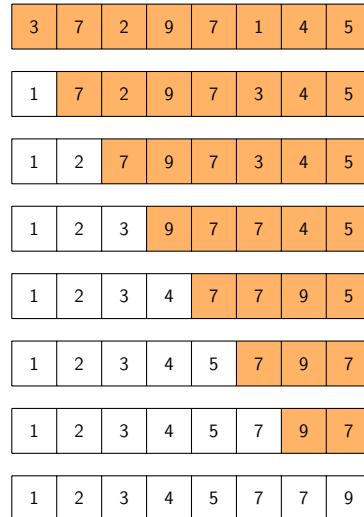


Selectionsort: Informell

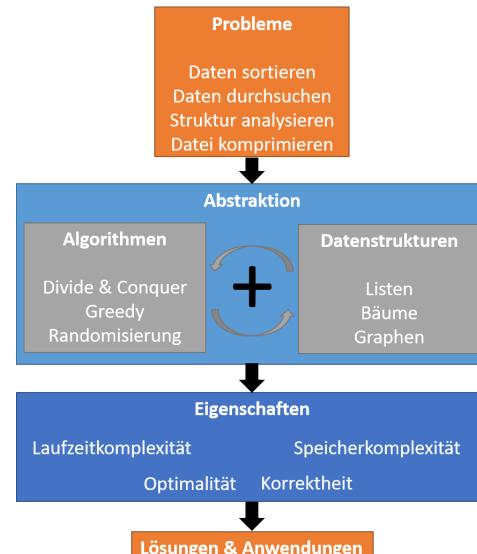


- ▶ Finde kleinstes Element an Positionen $0, \dots, n - 1$ und tausche es an Position 0
- ▶ Finde kleinstes Element an Positionen $1, \dots, n - 1$ und tausche es an Position 1
- ▶ ...
- ▶ Finde kleinstes Element an Positionen $n - 2, \dots, n - 1$ und tausche es an Position $n - 2$

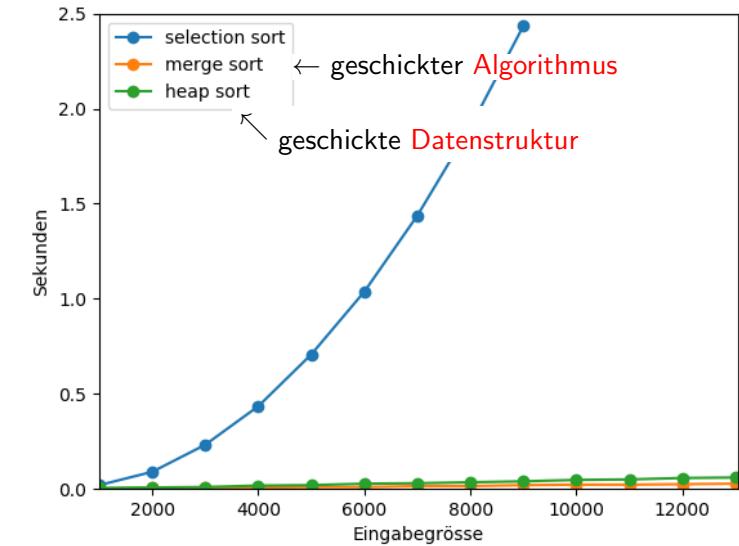
Selectionsort: Beispiel



Der Kurs Algorithmen und Datenstrukturen



Sortieralgorithmen: Laufzeit



Inhalt dieser Veranstaltung

