

# Algorithmen und Datenstrukturen

## Vorlesung #00 – Organisatorisches



Benjamin Blankertz

Lehrstuhl für Neurotechnologie, TU Berlin

[benjamin.blankertz@tu-berlin.de](mailto:benjamin.blankertz@tu-berlin.de)

18 · Apr · 2023



# Willkommen zu “Algorithmen und Datenstrukturen”

## **Algorithmen**

Verfahren zum  
Lösen von Problemen  
als schrittweise  
Anleitung

**Implementation durch Computerprogramm:  
Objekt-orientierte Programmierung in Java**

## **Datenstrukturen**

Organisation von  
Daten zur effizienten  
Verarbeitung

# Willkommen zu “Algorithmen und Datenstrukturen”

- ▶ AlgoDat Kenntnisse sind relevant:
- ▶ Sie helfen der Nachhaltigkeit!
- ▶ Tests bei Bewerbungsgesprächen
- ▶ Entscheidend: algorithmische Lösungsstrategien zu unbekannten Problemvarianten
- ▶ Ergebnisse für **Java** in der Umfrage von *hired.com* in der Rubrik 2019 State of Software Engineers – The Hottest Coding Languages:
- ▶ *Which programming languages do you primarily work with?*  
2. Platz (nach JavaScript)
- ▶ *Most Loved Programming Languages:*  
3. Platz (nach Python und JavaScript)
- ▶ *Most Hated Programming Languages:*  
2. Platz (nach PHP)

# Die Wichtigkeit von Laufzeitanalyse und -optimierung

- ▶ Diese Vorlesung erklärt, wie komplexe Probleme **effizient** gelöst werden können.
- ▶ **Gedankenexperiment**
- ▶ Laufzeit 1 min bei Eingabegröße  $N = 1000$
- ▶ Was passiert, wenn die Datenmenge wächst?
- ▶ Hierfür ist die Kenntnis der Wachstumsordnung der Laufzeit relevant.

## Die Wichtigkeit von Laufzeitanalyse und -optimierung (2)

- ▶ Annahme: Die Laufzeit für eine Eingabe der Größe  $N = 1000$  ist eine Minute.
- ▶ Laufzeit für Eingabegrößen  $10N$ ,  $100N$  und  $1000N$  in Abhängigkeit von der Wachstumsordnung:

Wachstum	W-Ordnung	$N = 1000$	$10N$	$100N$	$1000N$
konstant	1	1 Minute	1 Minute	1 Minute	1 Minute
logarithmisch	$\log N$	1 Minute	$1, \bar{3}$ Minuten	$1, \bar{6}$ Minuten	2 Minuten
linear	$N$	1 Minute	10 Minuten	2 Stunden	1 Tag
leicht überlinear	$N \log N$	1 Minute	13 Minuten	3 Stunden	$1 \frac{1}{2}$ Tage
quadratisch	$N^2$	1 Minute	2 Stunden	5 Tage	23 Wochen
kubisch	$N^3$	1 Minute	1 Tag	2 Jahre	2000 Jahre
exponentiell	$2^N$	1 Minute	inf	inf	inf

Der konstante Faktor in den Laufzeitfunktionen ist so gewählt, dass sich für  $N = 1000$  eine Minute ergibt. Die angegebenen Laufzeiten für  $10N$ ,  $100N$ ,  $1000N$  sind gerundet.

**Dozent:** Prof. Dr. Benjamin Blankertz

`<benjamin.blankertz@tu-berlin.de>`

**Wissenschaftliche Mitarbeitende:** Vera Röhr, Noah Schlegel

`<algodat@neuro.tu-berlin.de>`

## Tutorinnen

Jonas Kriegs

Nikolai Skutsch

Johannes Hack

Rudolf Sotnikow

Theresa Tratzmüller

Richard Meier zu Ummeln

Max Krüger

Mark Bremer

Petro Novotnyy

Jessica Priebe

Andreas Kreppold

John Cheung

Jan Schlenzka

Maximilian Schreff

Philipp Trilk

`<algodat.tutors@neuro.tu-berlin.de>`

**Fachmentor:**

Maath Oudeh

## Kurzvorstellung – Benjamin Blankertz

- ▶ Studium: Mathematik mit Nebenfächern Informatik und Mathematische Logik
- ▶ Seit 2012 Professor an der TU Berlin
- ▶ Forschungsschwerpunkt Neurotechnologie mit Methoden des Maschinellen Lernens
- ▶ Speziell: *Brain-Computer Interfaces* (BCI)

- ▶ **Benjamin Blankertz**
- ▶ Fachgebiet Neurotechnologie, Fakultät IV der TU Berlin
- ▶ Sprechstunde: Do 12:00 bis 13:00 Uhr, Raum 4.041
- ▶ Anmeldung über Sekretariat MAR 4-3:
- ▶ `<contact@neuro.tu-berlin.de>`



- ▶ **Vera Röhr**
- ▶ **Noah Schlegel**
- ▶ Kontakt: `algodat@neuro.tu-berlin.de`
- ▶ Kontakt zu den Tutorinnen: `algodat.tutors@neuro.tu-berlin.de`

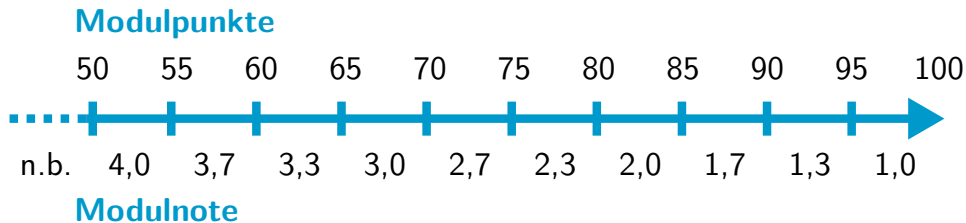
# Das Wichtigste für die Organisation

- ▶ Wichtigste Quelle für Informationen: ISIS Kurs “Algorithmen und Datenstrukturen”:  
<https://isis.tu-berlin.de/course/view.php?id=31624>
  - ▶ Skript, Videos, Übungsblätter, ...
  - ▶ Aktuelle Ankündigung
  - ▶ Termine, Räume (Tutorien, Klausuren, ...)
  - ▶ Diskussionsforen
  - ▶ Kontakte und Antworten auf häufige Fragen
- ▶ Melden Sie sich **möglichst bald** an sofern nicht schon geschehen
- ▶ Achten Sie auf ISIS Ankündigungen!

Ohne TU Account: “Als Gast anmelden” und dann als Gastschlüssel *AlgoDat23* eingeben.

# Modul “Algorithmen und Datenstrukturen”

- ▶ Modul “Algorithmen und Datenstrukturen”: MOSES #40022.
- ▶ Anmeldung zur Modulprüfung in MOSES **am besten direkt**. Anmeldungen und Abmeldungen sind bis **25.05.2023** möglich.
- ▶ Die Prüfungselemente des Moduls sind (Portfolioprüfung):
- ▶ **50% Übungen**: 10 Aufgabenblätter, je 5 Modulpunkte
- ▶ **50% Schriftlicher Test**: 60 Minuten, 50 Modulpunkte
- ▶ Benotung nach Notenschlüssel 2 der Fakultät IV:



- ▶ **als Pflicht, Wahlpflicht und freie Wahl:**
  - ▶ Anmeldung über MOSES, siehe vorige Seite
- ▶ **für Nebenhörerinnen und Gasthörerinnen:**
  - ▶ Anmeldung über Formular, siehe ISIS  
siehe auch <https://www.tu.berlin/go1343/>
- ▶ **im Orientierungsstudium MINTgrün oder Erasmus:**
  - ▶ Anmeldung per Email an `<contact@neuro.tu-berlin.de>`, siehe ISIS

- ▶ **Vorlesung**
- ▶ **Material** auf ISIS
  - ▶ Skript, Videos zum Vorlesungsstoff
  - ▶ Videos zu jedem Aufgabenblatt
- ▶ **Übungsaufgaben**
  - ▶ Programmieraufgaben
  - ▶ In Einzelabgabe
- ▶ **Rechnerübungen** in Rechnerräumen
- ▶ **Online-Tutorien**
- ▶ **Fachmentorien**

- ▶ Alle Aufgaben sind Programmieraufgaben mit Einzelabgabe.
- ▶ Aufgabenzettel gibt es jeweils am **Dienstag**.
- ▶ Abgabe in der übernächsten Woche am **Montag bis 23:59 Uhr**.
- ▶ Heute 'Blatt 0', mit Infos und Testabgabe

## **Bitte teilnehmen!**

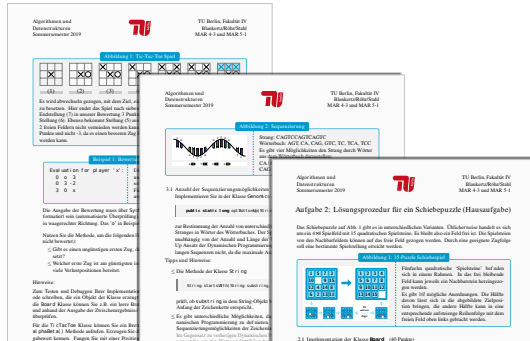
- ▶ In den folgenden Wochen gibt es zehn Aufgabenzettel, die für die Modulnote zählen.

# Übungsaufgaben

- ▶ Alle Aufgaben sind Programmieraufgaben mit Einzelabgabe.
- ▶ Aufgabenzettel gibt es jeweils am **Dienstag**.
- ▶ Abgabe in der übernächsten Woche am **Montag bis 23:59 Uhr**.
- ▶ Heute 'Blatt 0', mit Infos und Testabgabe

**Bitte teilnehmen!**

- ▶ In den folgenden Wochen gibt es zehn Aufgabenzettel, die für die Modulnote zählen.

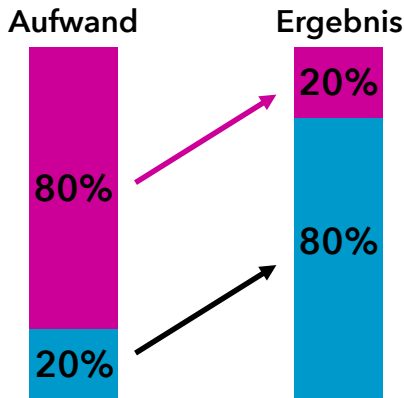


- ▶ Die Abgabe geschieht über ein Versionsverwaltungssystem
- ▶ Wir verwenden GIT, siehe heutiges Blatt
- ▶ Die Abgaben werden automatisch über spezielle Tests bewertet.
- ▶ Zwei Tests pro Tag. Einzelne Ausfälle sind möglich. Diese Bewertung kann zur Kontrolle benutzt werden.
- ▶ Das wichtigste Mittel zur Verbesserung der Implementationen sollte sorgfältiges Debugging und ggf. (selbstgeschriebene) Tests sein.
- ▶ Die letzte Einsendung vor dem Abgabetermin zählt. Endredaktion der Korrektur durch WiMis; die Punktzahl kann vom automatischen Testergebnis abweichen.



- ▶ Für Erfolg und das richtige Verständnis: Üben, Üben, Üben
- ▶ Es gibt viele Angebote zur Unterstützung beim Programmieren: nutzt sie!
- ▶ Macht die Programmierung der Übungsaufgaben **selbstständig**:
- ▶ Kein Quelltext von anderen Personen oder Quellen (z. B. Internet).
- ▶ Vorabprüfung mit Plagiatserkennungssoftware; Prüfung durch WiMis.
- ▶ Zu **Plagiaten** siehe Merkblatt Plagiate unserer Fakultät
- ▶ Diskussionen über Lösungswege sind **ausdrücklich erlaubt**,  
die Implementierung muss selbstständig gemacht werden.

# Bei der Bearbeitung beachten: Das Pareto Prinzip



## Regel in AlgoDat:

- ▶ Bereits 80/100 Punkte ergeben die vollen 5 Modulpunkte, die pro Blatt erreicht werden können.
- ▶ Der Rest ist kür und ergibt keine weiteren Modulpunkte.
- ▶ Also insgesamt: HA-Modulpunkte =  $\frac{50}{800} \sum_{k=1}^{10} \min(80, \text{Blatt}[i])$

- ▶ Es sind jeweils mindestens zwei Tutorinnen anwesend.
- ▶ Raum TEL 106 li + re
  - ▶ Mo 12:00 – 13:30 Uhr
  - ▶ Di 08:00 – 09:30 Uhr und 14:00 – 15:30 Uhr
  - ▶ Mi 14:00 – 15:30 Uhr
  - ▶ Fr 14:00 – 15:30 Uhr
- ▶ Anfang ab dieser Woche Mittwoch
- ▶ Unterstützung beim Programmieren
- ▶ wenige Plätze: bitte mit konkreten Fragen kommen  
nicht für das generelle Bearbeiten der Programmieraufgaben

- ▶ Hilfe für internationale Studierende:
  - ▶ Orientierung am Studienanfang
  - ▶ Fachliche Unterstützung bezüglich Inhalten von Vorlesung und Aufgaben
  - ▶ Vorbereitung auf Klausuren und Prüfungen
- ▶ Maath Oudeh
  - ▶ Mo 14-16 Uhr, Raum MAR 0.013
  - ▶ Do 14-16 Uhr, Raum MAR 0.015
  - ▶ Beginn Do 20.04.
  - ▶ Informationen im ISIS Kurs *EECS-Fachmentoring*:
  - ▶ <https://isis.tu-berlin.de/course/view.php?id=660>

# Überblick über den Kurs (momentaner Planungsstand)

1. Einführung Java: Objektorientierte Programmierung
2. Einführung Java: Vererbung, Generics, Debugging; Wachstumsordnungen
3. Einführung Java: Ausnahmenbehandlung, Unit Tests; Vorrangwarteschlangen
4. *Backtracking*, P und NP, Gierige Algorithmen
5. *Branch-and-Bound*, Alpha-Beta Suche, Randomisierte Algorithmen
6. Dynamische Programmierung
7. Graphenalgorithmen: Tiefen- und Breitensuche
8. Minimale Spannbäume
9. Kürzeste Wege, Topologische Sortierung
10. Heuristische Algorithmen, Approximative Algorithmen
11. Effektive Symboltabellen mit *Hashing*
12. Flussgraphen, maximaler Fluss, minimaler Schnitt

## ▶ **Vorlesungen**

- ▶ Einführung in die Programmierung (*IntroProg*)
- ▶ Rechnerorganisation

## ▶ **Kenntnisse**

- ▶ Elementare Datenstrukturen: verkettete Listen, Arrays, Bäume, binäre Halden
- ▶ Komplexitätsanalyse von Algorithmen
- ▶ Sortier- und Suchverfahren
- ▶ Aufbau und Funktionsweise eines Computers

## ▶ **Fähigkeiten**

- ▶ Imperative Programmierung am Beispiel C
- ▶ Versionskontrollsysteme mit svn oder git

## Kenntnis

- ▶ von komplexere Datenstrukturen (Graphen, Hashtabellen, ...)
- ▶ der wichtigsten elementaren Algorithmen (kürzeste Wege, minimaler Spannbaum, Hashing, ...)
- ▶ der Verfahren zur Entwicklung effizienter Algorithmen (Dynamisches Programmieren, *Branch-and-Bound*, Alpha-Beta Suche, Heuristische Verfahren, ...)

## Fähigkeit

- ▶ für ein Anwendungsproblem passende Algorithmen und Datenstrukturen zu identifizieren
- ▶ Rechenzeit und Speicherbedarf von Algorithmen abzuschätzen
- ▶ verständliche Programme in Java zu schreiben
- ▶ ... unter Verwendung von Objektorientierung und
- ▶ ... mit Debugging, Unit Tests und Kommentierstandards

## Programmierung in Java

- ▶ Ullmann C, **Java ist auch eine Insel**. 13. Auflage, Rheinwerk Computing, 2018. Onlinefassung: <http://openbook.rheinwerk-verlag.de/javainsel>
- ▶ Programming Guide – Java: <https://programming.guide/java/>
- ▶ Sedgewick R & Wayne K, **Introduction to Programming in Java: An Interdisciplinary Approach**. 2. Auflage, Addison-Wesley Professional, 2017. Onlinefassung: <https://introcs.cs.princeton.edu/java>
- ▶ MOOC Platform Hyperskill: <https://hyperskill.org>

## Algorithmen und Datenstrukturen

- ▶ Sedgewick R & Wayne K, **Algorithmen: Algorithmen und Datenstrukturen**. Pearson Studium, 4. Auflage, 2014. ISBN: 978-3868941845
- ▶ Ottmann T & Widmayer P **Algorithmen und Datenstrukturen**. Springer Verlag, 5. Auflage; 2011. ISBN: 978-3827428042
- ▶ Cormen TH, Leiserson CE, Rivest R, Stein C, **Algorithmen - Eine Einführung**. De Gruyter Oldenbourg, 4. Auflage; 2013. ISBN: 978-3486748611