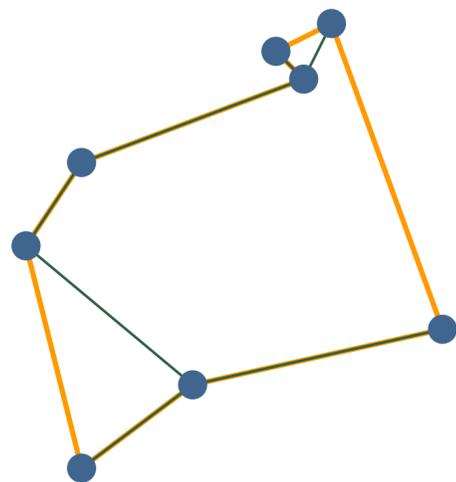
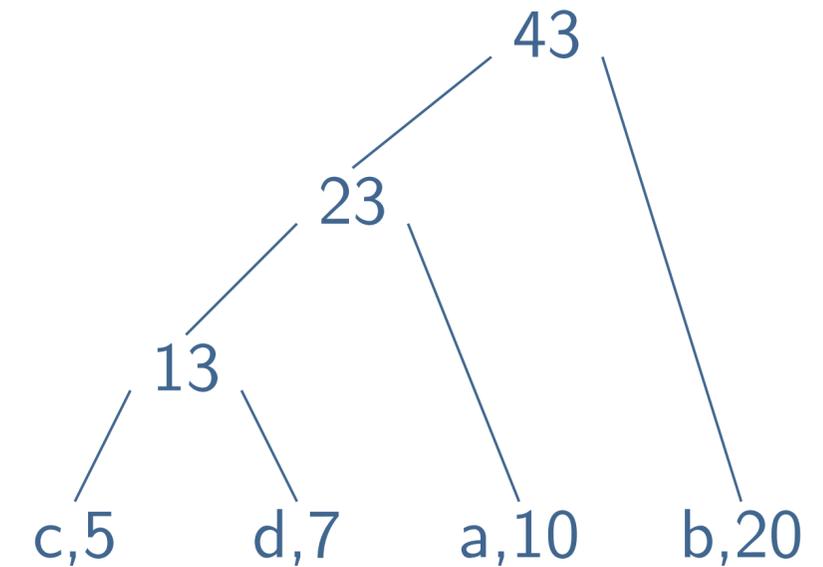
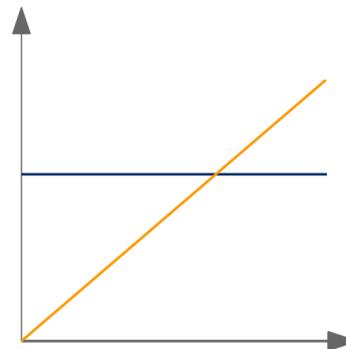
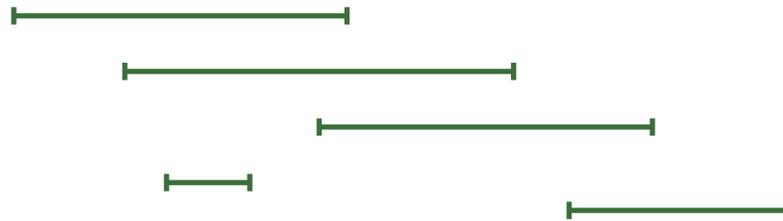


# Proseminar Theoretische Informatik

aabaababa  
bab



$$AB = A_h B_h b^{2n} + (A_h B_l + A_l B_h) b^n + A_l B_l$$

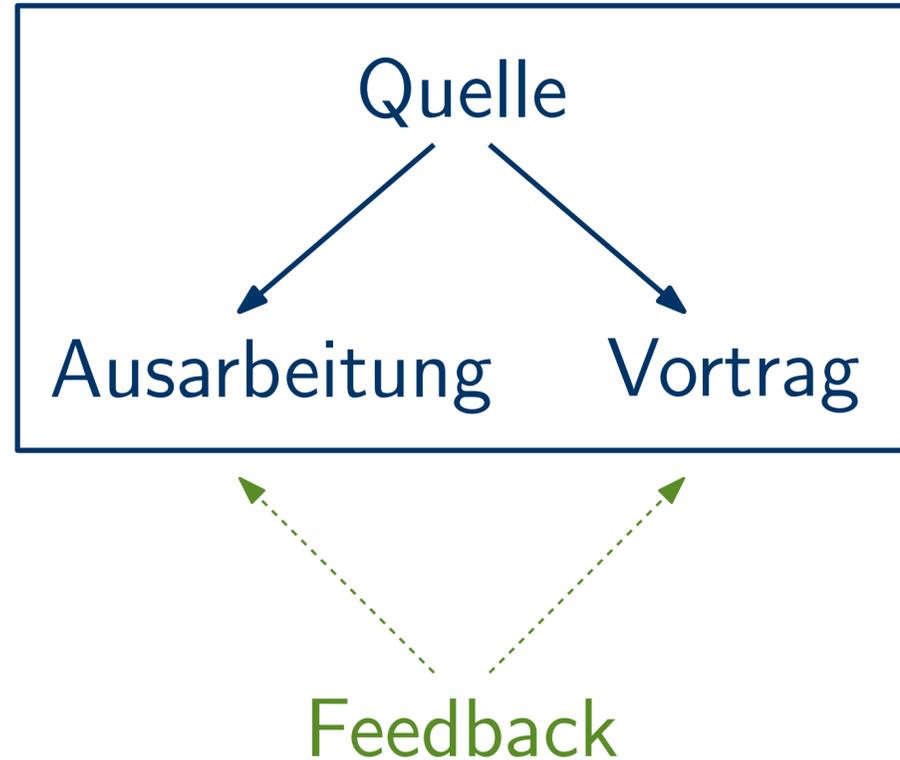


Katharina Klost

Sommersemester 2020

# Einleitung

## Ziele:



## Umsetzung:

Woche 1:	<b>Heute!</b> Einführung und Themenvergabe	
Woche 2:	Einarbeiten in Themen Ressourcen für Vorträge	Freiwilliger Termin Demo Ipe
Woche 3:	Kurze Themenvorstellung 2-5 Minuten pro Thema	
Wochen 4-5:	Aufschreiben der Ausarbeitungen	Keine Präsenz!
Wochen 6-7:	Peer Review der Ausarbeitungen Fertigstellen der Ausarbeitungen	Keine Präsenz!
Wochen 8-13:	Vorträge	

Jederzeit: Rückfragen per Mail - [k.klost@fu-berlin.de](mailto:k.klost@fu-berlin.de)

# Organisatorisches

**Präsenztermine**      WebEx - Meetings      Link in Whiteboard

**Sprechstunde**      Dienstags 14:00 - 15:00

Nach dem Seminar für kurze Rückfragen

<https://fu-berlin.webex.com/meet/k.klost>

**Persönliche Vorbesprechung:**

≈ 1 Woche vor Vortrag

Fertiges Vortragskonzept + Folienentwurf

**Regelmäßige und aktive Teilnahme:**

Ausarbeitung

Peer Review

Feedback

# Ausarbeitung und Vortrag

## Ausarbeitung

### Allgemeines:

Template im Whiteboard  
Maximal 270 Zeilen

### Struktur:

Einleitung  
Hauptteil  
Platz für ausführlichere Beweise  
Schluss

Mehr Hinweise im Template

## Vortrag:

### Allgemeines:

Folienvortrag   
"live" in Webex

30-40 Minuten + Fragen

Feedback für jeden Vortrag im Whiteboard

### Struktur:

Klare Trennung Einleitung - Hauptteil - Schluss  
anschauliche Beispiele  
Bilder Bilder Bilder

# Details zur Bewertung

## Struktur

Einleitung    Definition + Einordnung des Problems



Hauptteil    Wesentliche Ideen



Schluss    klares Fazit

## Vortragstechnik

Frei gehalten?

Im Zeitlimit und passendes Tempo?

Passender Einsatz von Medien?

Folien klar gestaltet und lesbar?

Bilder?

## Inhalt

Tiefe und Breite?

Bezüge zu Bekanntem?

Fokus auf wichtige Elemente?

Beispiele?

Definitionen und Beweise nachvollziehbar?

## Gesamteindruck

Verständlich?

Gut vorbereitet?

Interessante Gestaltung?

Stoff durchdrungen?

Lösung von schwierigen Situationen?

# Themen

## Datenkompression

Kompression von Text

- Huffman-Code
- LZW

[Erickson, Algorithms, 4.4]

[Cormen, Algorithms Unlocked, 9]

## Textsuche

Amirhossein Rajabi Pour

Wie können wir einen Teiltext schnell suchen?

- Naiver Algorithmus
- Rabin-Karp Algorithmus
- Knuth-Morris-Pratt Algorithmus

[Cormen et al., Algorithmen - eine Einführung, Kap 32]

## Karatsuba-Multiplikation

Qianli Wang

Wie können wir mittels Divide und Conquer schnell multiplizieren?

[Erickson, Algorithms, Kap 1.10]

[Vöcking et al., Taschenbuch der Algorithmen, Kap 11]

## Online-Algorithmen

Wie können wir entscheidungen treffen, wenn die Zukunft unklar ist?

[Vöcking et al., Taschenbuch der Algorithmen, Kap 39,40]

# Themen

## Das Problem des Handelsreisenden

Wie weit kann ein schweres Problem gelöst werden?

- Approximation für metrisches TSP
- Dynamische Programmierung für genaues Ergebnis

[Vöcking et al., Taschenbuch der Algorithmen, Kap 42]

## Stabile Paarung

Wie findet man eine Paarung, so das keiner Unglücklich ist?

[Erickson, Algorithms, Kap 4.5]

## Dynamisches Programmieren Nazar Sopiha

Weitere Anwendungen des DP Paradigmas

- Teilsummen Problem
- Optimale Binäre Suchbäume
- Unabhängige Knotenmenge auf Bäumen

[Erickson, Algorithms, Kap 3.8 - 3.9]

## Gierige Algorithmen

Wenn gier gut funktioniert:

- Magnetspeicher
- Kursplanung

[Erickson, Algorithms, Kap 4.1 - 4.3]

# Themen

## Page Rank

Der große Durchbruch von Google

[Vöcking et al., Taschenbuch der Algorithmen, Kap 10]

[Ausiello + Rossella, The Power of Algorithms, Kap 5]

## Flussnetzwerke - Einführung Fatima Azham

Eine Möglichkeit zum Lösen von Optimierungsproblemen

[Vöcking et al., Taschenbuch der Algorithmen, Kap 36]

[Cormen et al., Algorithmen - eine Einführung, Kap 26]

## Kleinster Umschließender Kreis Ragna Dalewski

Wie kann die Feuerwehr schnell alle erreichen?

[Vöcking et al., Taschenbuch der Algorithmen, Kap 38]

## Streckenschnitt Paul Weiß

Wie finden wir effizient alle Schnittpunkte einer Menge von Strecken?

[Cormen et al., Algorithmen - eine Einführung, Kap 33]

[Berg et al., Computational Geometry, Kap 2]

# Zusammenfassung

Thema → Überblick → Ausarbeitung → Vortrag

Fragen?