

Auch Klausurstoff (kann noch geändert werden)

Neberst - Übungen

① Grundlegende Begriffe; Bsp: - Baum: \_\_\_\_\_ Graph, Eigenschaften etc.

- Höhe  $h$ : Baum hat mind. \_\_\_\_\_ und höchster \_\_\_\_\_ Knoten.

Hilfreich: Liste von Definitionen:

Begriff	Definition
ungerichteter Graph	$G = (V, E)$ , Vollst. Menge $E \subset V \times V$ (keine Erklärung)

② Sortieren:  
ein Algorithmus <sup>(bekannt)</sup> steht da.

- Korrektheit beweisen durch finden einer Invariante + Induktion

③ Asymptotik

-  $o, O, \omega, \Omega, \Theta$  einordnen können

$f$   $\uparrow$  also  $f \in ?(g)$   
 $\log n$   $n$

④ Rekurrenzgleichung

Mastertheorem anwenden

$$T(n) = T\left(\frac{n}{2}\right) + 1$$

Mastertheorem besagt siehe  $a = \dots$   $b = \dots$  usw. Ergebnis

⑤ Sortieren [wird evtl. ersetzt]

- unvollständiger Pseudo-Code ergänzen (Merge sort o.ä.)

(Code Rahmen gegeben)

- Algorithmus auf einfaches Beispiel anwenden.  $\rightarrow$  Darstellung z.B. durch Rekursionsbaum

⑥ Binärer Suchbaum

- Elementare Begriffe (Und, Eltern etc.)

$\rightarrow$  Einfügen / Entfernen / Suchen (Algorithmen kennen)

⑦ Tiefensuche und topologische Sortierung oder Breitensuche

- Algorithmen kennen Discovery Time / Finishing Time etc.

[Vorbereitung: Beispiele durchrechnen, wie funktioniert der Algorithmus im Prinzip]

## ⑧ Hashkette

- verkettete Liste
- offene Adressierung

gegeben: Schlüssel 13, 28, 16, 9, 14, 43

Hashfunktion  $h(x) = x \bmod 3$

erwartet: Hashkette

## ⑨ Max. Flusse

Flussnetzwerk  $\xrightarrow{\text{alg.}}$  max Fluss finden  
verifizieren

## ⑩ B-Bäume

Gegeben: B-Baum, minimalgrad  $t$

- Operationen kennen: einfügen, aktualisieren, löschen etc.

## ⑪ Minimal aufspannende Bäume

gegeben: Graph, Startknoten, Gewichte

Prim anwenden

Altkoll nicht vorhanden (ändert sich nah):

- dynamische Programmierung
  - Greedy
  - Pattern Matching
  - NP-Vollständigkeit
- } 2 davon werden mit  
2 von birk ex. Aufgaben ersetzt.

- Algorithmen kennen und verstanden haben!