

Aufgabe 1: 10 Punkte

- a) Geben Sie eine formale Spezifikation für den ADT Sequenz mit mind. 4 Funktionen an.
- b) Nenne eine andere Implementierung des ADT Sequenz als einfach verkettete Listen
- c) Implementieren Sie die Funktion last für einfach verkettete Listen in Pseudocode
Eingabe: eine nicht leere Liste
Ausgabe: Wert des letzten Listenelements
→ Tipp: Verwenden Sie die ADT zu Listenelement und Zeiger, welche aus der Vorlesung bekannt sind.

Aufgabe 2: 8 Punkte

Adjazenzliste eines ungerichteten Graphen G gegeben:

A	B,C
B	A,D
C	A,D,E
D	B,C,E
E	C,D

- a) Warum ist G kein Baum? Erklären anhand der Definition von Bäumen.
- b) Fertigen Sie eine Skizze von H an, welcher ein aufspannender Baum von G. A ist die Wurzel und ist adjazent zu B und C
- c) Traversieren Sie H nach
 - a) Preorder
 - b) Postorder→ Falls ein Knoten nur ein Kind hat, wird dieses als linkes Kind behandelt
- d) Wie tief ist ein vollständiger Binärbaum mit 64 Blättern?
- e) Wieviele Blätter hat ein fast vollständiger Binärbaum der Tiefe n
 - a) mindestens?
 - b) höchstens ?

Aufgabe 3: 12 Punkte

Gegeben: $t(n) = 8 \cdot t(n/4) + n^2$ $t(1) = 10$

- a) Geben Sie die Komplexität dieser Rekursionsgleichung.
- b) Entwerfen Sie einen Algorithmus nach dem Prinzip „Teile und Herrsche“, welcher eine Liste von ganzen Zahlen übergeben bekommt und die Summer aller Elemente daraus berechnet. (In Python)
- c) Was ist der worst-case des Algorithmus aus b) ? Geben Sie ihren Lösungsweg an.
- d) [Aussagen zu Dynamischer Programmierung zum ankreuzen]

Aufgabe 4: 12 Punkte

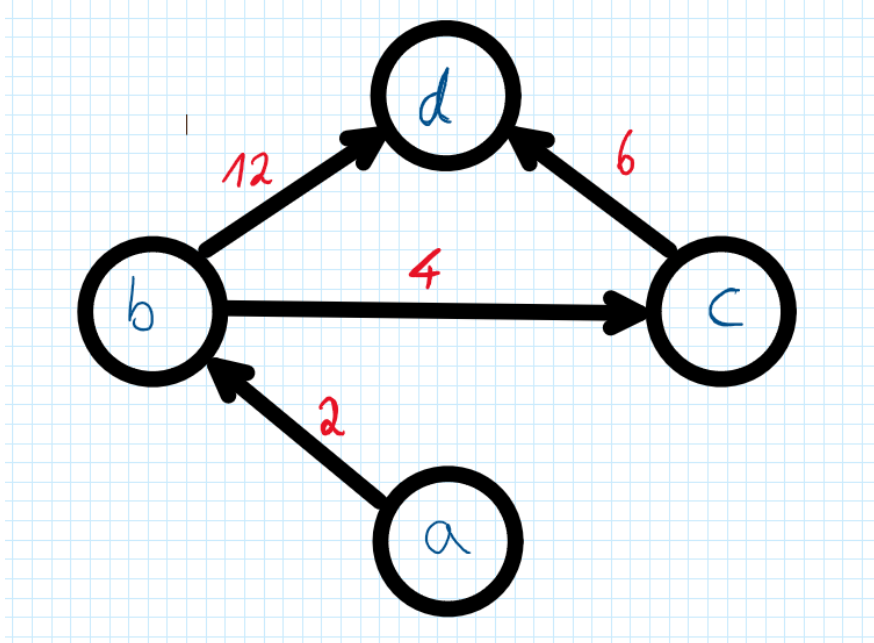
- a) Welche Voraussetzungen muss eine Folge erfüllen, damit binäre Suche anwendbar ist?
- b) $A = \{11, 7, 3, 8, 1, 2\}$
Geben Sie den binären Suchbaum an, der sich ergibt wenn man die Elemente nacheinander dem Baum hinzufügt.
- c) A als AVL-Baum darstellen, inklusive Balancefaktoren aller Knoten. Geben Sie den Baum nach jedem Hinzufügen eines Elements und gegebenenfalls balancieren an.

Aufgabe 5: 10 Punkte

	Worst-case	Average-case	In-place? Ja/Nein
Insertionsort			
Mergesort			
Quicksort			
Heapsort			

Aufgabe 6: 10 Punkte

- a) Geben Sie die formale Definition der Austauscheigenschaft von Matroiden an.
b) Welchen Vorteil hat man, wenn man weiß, dass die Lösung des kanonische Greedy-Algorithmus ein Matroid ist?
c) Dijkstra-Algorithmus mit dem Knoten a als Startwert.
Geben Sie alle im Algorithmus gespeicherten Werte zu jeder Iteration an.

**Aufgabe 7: 11 Punkte**

M & N sind Mengen, A ist eine Teilmenge von M & B ist eine Teilmenge von N

Q:

Eingabe: x Element M

Ausgabe: 1 falls x Element A, ansonsten 0

R:

Eingabe: x Element N

Ausgabe: 1 falls x Element B, ansonsten 0

- a) Definieren Sie formal Transformation von Q auf R.
b) Definieren Sie, wann ein Problem NP-vollständig heißt.
c) Sind NP-vollständige Probleme die schwersten oder leichtesten Probleme von NP?
d) Wie kann man durch Konstruktion eines einzigen, speziellen Algorithmus beweisen, dass P=NP gilt?

e) Begründen Sie durch Entwurf eines Algorithmus, dass das Hamilton-Pfad-Problem teil von NP ist.

Aufgabe 8: 7 Punkte

$$f(n) = (\text{Logarithmus } n \text{ zur Basis } 2)^2$$

$$g(n) = n$$

- a) Welche Relation hat $f(n)$ zu $g(n)$? Element $O/o/w/\Omega/\Theta$ ($g(n)$)?
b) Weisen Sie diese Relation nach.