

Probeklausur Algorithmen

(Prof. Kaufmann/Bekos/Schneck, Sommersemester 2018)

- Die Bearbeitungszeit beträgt **90 Minuten**, elektronische Hilfsmittel sind verboten, nur ein handbeschriebenes DIN-A4-Blatt mit Notizen ist erlaubt.
- Von den 6 Aufgaben mit jeweils 6 möglichen Punkten werden 5 bewertet. Geben Sie an, welche Aufgabe **nicht** bewertet werden soll.

Viel Erfolg!

Name:

Matr.-Nr.:

Studiengang:

Angestr. Abschluss:

	Punkte	max
1		6
2		6
3		6
4		6
5		6
6		6
Σ		30

Aufgabe 1: (Rekursion)

(6 Punkte)

Betrachten Sie folgende Rekursionsgleichung: Ist n eine Dreierpotenz, so ist

$$T(n) = 2 \cdot T(n/3) + 5n$$

$$T(1) = 5$$

Geben Sie eine geschlossene Form für $T(n)$ an und benutzen Sie dabei **nicht** das Mastertheorem. Beweisen Sie die Korrektheit Ihrer Antwort.

Aufgabe 2: (Dynamische Programmierung)

(6 Punkte)

Betrachten Sie folgende Variante des Rucksack-Problems: Gegeben seien n Objekte o_1, \dots, o_n . Jedes Objekt o_j ($1 \leq j \leq n$) hat ein Gewicht w_j . Nun soll ein Rucksack mit einer **minimalen** Anzahl an Objekten gefüllt werden, so dass das Gesamtgewicht des Rucksacks genau W ist. Gehen Sie davon aus, dass w_1, \dots, w_n und W natürliche Zahlen größer 0 sind.

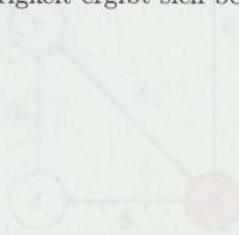
- a) Geben Sie einen Algorithmus in Psuedocode an, der die minimale Anzahl an Elementen im Rucksack berechnet. Verwenden Sie dabei **Dynamische Programmierung**.
- b) Begründen Sie die Korrektheit Ihres Programms.
- c) Geben Sie die Laufzeit Ihres Programms an.

Aufgabe 3: (Suchbäume)

(6 Punkte)

Wir betrachten eine spezielle Klasse von Bäumen, die wir *1-3-Bäume* nennen. Diese Bäume mit Wurzel haben die Eigenschaft, dass jeder innere Knoten entweder 1 oder 3 Kinder hat, aber **niemals** 2.

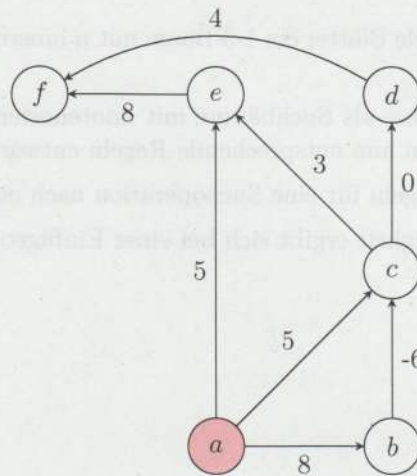
- Geben Sie an, wie viele Blätter ein 1-3-Baum mit n inneren Knoten mindestens und höchstens haben kann.
- Wir wollen 1-3-Bäume als Suchbäume mit knotenorientierter Speicherung benutzen. Für 1-3-Suchbäume sollen nun entsprechende Regeln entworfen werden.
 - Wie sind die Regeln für eine Suchoperation nach einem Schlüssel x ?
 - Welche Schwierigkeit ergibt sich bei einer Einfügeoperation und wie könnten Sie diese beseitigen?



Aufgabe 4: (Billigste Wege)

(6 Punkte)

Betrachten Sie den unten dargestellten **gerichteten** Graphen $G = (V, E)$ mit Kantenkosten $c: E \rightarrow \mathbb{Z}$.



Führen Sie den Dijkstra Algorithmus auf G aus, gestartet am Knoten a .

- a) Tragen Sie dazu die entsprechenden Paare in folgende Tabelle ein. Für einen Knoten $v \in V$ ist dabei $d'(v)$ die vorläufig berechnete Distanz und $d(v)$ die endgültige Distanz von Knoten a zu Knoten v . Spalte S enthält den zuletzt zur Lösung hinzugenommenen Knoten v zusammen mit $d(v)$. Spalte S' enthält die Knoten v , die Kandidaten für die nächste Iteration sind, jeweils zusammen mit $d'(v)$.

Schritt	$S: (v, d(v))$	$S': (v, d'(v))$
1	$(a, 0)$	
2		
3		
4		
5		
6		

- b) Für welche Knoten berechnet der Dijkstra-Algorithmus eine **falsche** kürzeste Distanz?
 c) Warum? Wie sind die richtigen Distanzen?
 d) Gibt es einen Algorithmus, der dieses Problem umgeht? Welcher ist das?

Aufgabe 5: (Hashing)

(6 Punkte)

Sei $B = \{0, 1\}^*$ die Menge aller endlichen Bitstrings und $T = \{x \in \mathbb{Q} \mid 0 \leq x \leq 1\}$ das rationale Intervall zwischen 0 und 1. Betrachten Sie folgende Funktion:

$$\iota: B \rightarrow T,$$

$$\iota(w_1 \cdots w_n) := \sum_{i=1}^n w_i \cdot 2^{-i}$$

Diese Funktion bildet also einen Bitstring $w_1 \cdots w_n$ auf die binäre Fließkommazahl $0, w_1 \cdots w_n$ ab. Seien außerdem für jedes $m \in \mathbb{N}$

$$h_m: B \rightarrow T$$

$$h_m(w_1 \cdots w_n) := \begin{cases} \iota(w_1 \cdots w_m) & m < n \\ \iota(w_1 \cdots w_n) & m \geq n \end{cases}$$

jeweils eine Hashfunktion.

- Berechnen Sie $\iota(101010)$. Berechnen Sie außerdem $\iota(1100)$, sowie $h_2(1100)$ und $h_{42}(1100)$.
- Geben Sie in Abhängigkeit von $m \in \mathbb{N}$ die Menge C_m aller Bitstrings an, die mit dem Bitstring $w = 101$ kollidieren.
- Beschreiben Sie $\iota(C_m) \subseteq T$ in Abhängigkeit von $m \in \mathbb{N}$.

Aufgabe 6: (Tiefensuche und transitive Hülle)

(6 Punkte)

Sei $G = (V, E)$ ein ungerichteter Graph in Adjazenzlistendarstellung mit $|V| = n$ und $|E| = m$. Die transitive Hülle $G^* = (V, E^*)$ von G ist definiert durch

$$E^* = \{\{v, w\} \mid v \neq w \text{ und es gibt einen Pfad von } v \text{ nach } w \text{ in } G\}.$$

- Mit welchen Knoten ist ein Knoten $v \in V$ in der transitiven Hülle G^* verbunden?
- Beschreiben Sie, wie der Graph G^* der transitiven Hülle aussieht.
- Sei ein Knoten $v \in V$ gegeben. Zeigen Sie, dass Sie mit Hilfe der Tiefensuche in Zeit $O(n+m)$ die Adjazenzliste von v in G^* bestimmen können.