

Algorithmen und Datenstrukturen 1 (ADS)	Theoretische Einzelprüfung (1. Nachtermin)	25.01.2021		1
---	--	------------	--	---

	21		20		31		23		35		24		39		26
+		+		+		+		+		+		+		+	
	z_1		z_2		z_3		z_4		z_5		z_6		z_7		z_8

Aufgabe 1 [2]

Fügen Sie in obiger Tabelle in den leeren Kästchen, vor denen das Pluszeichen steht, die Ziffern Ihrer Matrikelnummer ein. Führen Sie die Additionen durch und ermitteln Sie die Zahlen z_1 bis z_8 .

Aufgabe 2 [18]

- a. [6] Für die Laufzeit T einer Funktion wurde die Rekurrenzgleichung

$$T(n) = z_3 * T\left(\frac{n}{3}\right) + 4n^2 + z_7n + 42 \text{ für } n \in \mathbb{N}$$

ermittelt. Bestimmen Sie mittels Master-Theorem das asymptotische Laufzeitverhalten der Funktion in Θ -Notation. Ersetzen Sie die Werte z_i durch die Ergebnisse aus Aufgabe 1.

- b. [6] Kann jede beliebige Rekurrenzgleichung mithilfe des Master-Theorems gelöst werden? Begründen Sie Ihre Antwort kurz.

- c. [6] Ermitteln Sie die asymptotische Laufzeitkomplexität des folgenden Codes in Θ -Notation:

```
int xpowy(int x, int n)
{
    if (n==0)
        return 1;
    if (n==1)
        return x;
    if ((n % 2) == 0)
        return xpowy(x*x, n/2);
    else
        return xpowy(x*x, n/2) * x;
}
```

Algorithmen und Datenstrukturen 1 (ADS)	Theoretische Einzelprüfung (1. Nachtermin)	25.01.2021		2
---	--	------------	--	---

Aufgabe 3 [20]

Die Werte z_1 bis z_8 . (aus Aufgabe 1) seien in dieser Reihenfolge von links nach rechts in einem Array gespeichert. Sortieren Sie die Werte aufsteigend mit

- [8] Quicksort
- [4] Counting Sort (**Achtung:** verwenden Sie $z_n \% 10$ als zu sortierende Werte!)
- [8] Heap Sort

Geben Sie alle notwendigen Schritte so genau an, dass die Arbeitsweise des jeweiligen Algorithmus klar ersichtlich wird.

Algorithmen und Datenstrukturen 1 (ADS)	Theoretische Einzelprüfung (1. Nachtermin)	25.01.2021		3
---	--	------------	--	---

Algorithmen und Datenstrukturen 1 (ADS)	Theoretische Einzelprüfung (1. Nachtermin)	25.01.2021		4
---	--	------------	--	---

Aufgabe 4 [20]

- a. [9] Fügen Sie die Werte $\mathbf{z_2}$ bis $\mathbf{z_8}$ aus Aufgabe 1 (in dieser Reihenfolge) in eine zu Beginn leere Hashtabelle der Länge 7 ein. Verwenden Sie als Hashfunktion $\mathbf{h(k) = k \% 7}$ und Double Hashing zur Kollisionsbehandlung. Die zweite Hashfunktion ist $\mathbf{g(k) = k \% 5 + 1}$.
Skizzieren Sie den Zustand der Hashtabelle nach jedem Einfügeschritt. (Anmerkung: Werte können mehrfach in der Hashtabelle gespeichert werden.)
- b. [1] Löschen Sie den Wert $\mathbf{z_5}$ aus der Tabelle und skizzieren Sie den Zustand der Hashtabelle.
- c. [5] Geben Sie den Kollisionspfad (besuchte Indexpositionen) bei einer Suche nach dem Wert $\mathbf{z_4}$ an.
- d. [5] Geben Sie den Kollisionspfad (besuchte Indexpositionen) bei einer Suche nach dem Wert 59 an.

Algorithmen und Datenstrukturen 1 (ADS)	Theoretische Einzelprüfung (1. Nachtermin)	25.01.2021		5
---	--	------------	--	---

Algorithmen und Datenstrukturen 1 (ADS)	Theoretische Einzelprüfung (1. Nachtermin)	25.01.2021		6
---	--	------------	--	---

Aufgabe 5 [20]

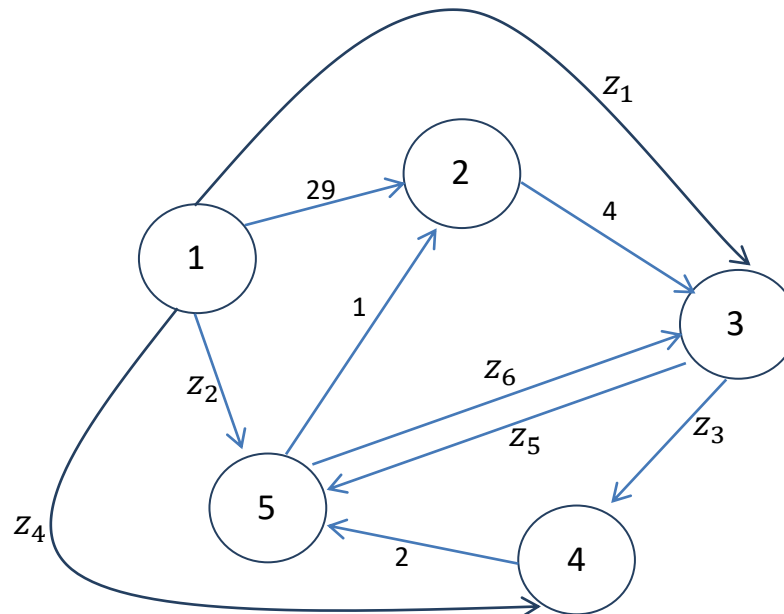
- a. [4] Fügen Sie die Werte z_2 bis z_8 aus Aufgabe 1 (in dieser Reihenfolge) in einen zu Beginn leeren, binären Suchbaum ein. Skizzieren Sie den Zustand des Baums nach jedem Einfügeschritt.
(Anmerkung: Werte können mehrfach im Baum gespeichert werden.)
- b. [4] Geben Sie in C++ ähnlicher Notation die Definition einer möglichen Datenstruktur für einen binären Suchbaum an.
- c. [8] Geben Sie in C++ ähnlicher Notation eine Definition einer Funktion oder Methode an, die das Maximum der im binären Suchbaum gespeicherten Werte ermittelt.
- d. [4] Bestimmen Sie die Laufzeitkomplexität Ihrer Funktion abhängig von der Anzahl n der im Suchbaum gespeicherten Werte in O-Notation. Begründen Sie Ihr Ergebnis kurz.

Algorithmen und Datenstrukturen 1 (ADS)	Theoretische Einzelprüfung (1. Nachtermin)	25.01.2021		7
---	--	------------	--	---

Aufgabe 6 [20]

Gegeben ist der folgende gerichtete Graph

(die Werte z_1 bis z_6 sind aus Aufgabe 1 zu übernehmen):



- [3] Geben Sie die Adjazenzmatrix des Graphen an.
- [3] Skizzieren Sie die Adjazenzliste des Graphen.
- [10] Bestimmen Sie mit dem Algorithmus von Kruskal einen minimal spannenden Baum des Schattens des Graphen. (Sie erhalten den Schatten des Graphen, indem Sie die Richtungen der Kanten vernachlässigen. Werden dann zwei Knoten durch zwei oder mehr Kanten verbunden, so werden diese Kanten zu einer zusammengefasst. Anders ausgedrückt: Zwei Knoten x und y im Schatten sind genau dann durch eine ungerichtete Kante verbunden, wenn im ursprünglich gerichteten Graphen zumindest eine der Kanten von x nach y oder von y nach x existiert. Als Gewicht der ungerichteten Kante wählen sie jeweils das Minimum aller durch sie repräsentierten gerichteten Kanten.) Notieren Sie alle Zwischenschritte so genau, dass klar ist, wann welche Kante zum spannenden Baum hinzugefügt wird.
- [2] Ist der oben dargestellte Graph topologisch sortierbar? Falls ja, geben Sie eine topologische Sortierung an, andernfalls begründen Sie, warum eine solche nicht gefunden werden kann.
- [2] Welche der folgenden Voraussetzungen ist hinreichend, damit der Dijkstra-Algorithmus das korrekte Resultat liefert? Zutreffendes bitte ankreuzen.

(1) Alle Kantengewichte des Eingabegraphen sind nicht-negativ.

(2) Der Eingabegraph ist ein DAG.

(3) Der Eingabegraph enthält keinen negativen Kreis.

(4) Der Eingabegraph enthält einen negativen Kreis.

Algorithmen und Datenstrukturen 1 (ADS)	Theoretische Einzelprüfung (1. Nachtermin)	25.01.2021		8
---	--	------------	--	---