

Algorithmen und Datenstrukturen 1 (ADS VU)	Theoretische Klausur	25.06.2019	Mnr.:	Name:	1
--	-------------------------	------------	-------	-------	---

13	11	32	15	18	23	39	34
+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>
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]

Gegeben sind folgende Funktionen:

```
void g(int i, int n) {
    if (i>0) {
        for (int j=n+10; j>0; j-=5)
            g(i-2, n);
    }
}
```

```
void f(int n) {
    if (!n) return;
    f(n/(z4%10+2));
    g(z6%5+1, n);
    for (int i=0; i<z7%10; i=i+2)
        f(n/(z4%10+2));
    g(z6%5+1, n);
}
```

Berechnen Sie die Laufzeit der Funktion f in Θ -Notation abhängig von n . Setzen Sie dazu für z_6 , z_7 und z_8 die in Aufgabe 1 ermittelten Werte ein. (Hinweis: Erstellen Sie Rekurrenzgleichungen für die Laufzeiten von g bzw. f und lösen Sie diese mittels fortgesetztem Einsetzen bzw. Master Theorem.)

Aufgabe 3 [20]

Die Werte z_1 bis z_8 (aus Aufgabe 1) jeweils modulo 10 (zB $z_1\%10$, $z_2\%10$, usf.) seien in dieser Reihenfolge von links nach rechts in einem Array gespeichert. Sortieren Sie die Werte aufsteigend mit

- [8] Quicksort
- [4] Selectionsort
- [8] Heapsort

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

Algorithmen und Datenstrukturen 1 (ADS VU)	Theoretische Klausur	25.06.2019	Mnr.:	Name:	2
--	-------------------------	------------	-------	-------	---

Aufgabe 4 [20]

- [9] Fügen Sie die Werte z_2 bis z_8 aus Aufgabe 1 (in dieser Reihenfolge) in eine zu Beginn leere Hashtabelle der Länge 7 ein. Verwenden Sie als Hashfunktion $h(k) = k \% 7$ und double hashing zur Kollisionsbehandlung. Die zweite Hashfunktion ist $g(k) = k \% 5 + 1$.
Skizzieren Sie den Zustand der Hashtabelle nach jedem Einfügeschritt.
- [1] Löschen Sie den Wert z_5 aus der Tabelle und skizzieren Sie den Zustand der Hashtabelle.
- [5] Geben Sie den Kollisionspfad (besuchte Indexpositionen) bei einer Suche nach dem Wert z_2 an.
- [5] Geben Sie den Kollisionspfad (besuchte Indexpositionen) bei einer Suche nach dem Wert 50 an.

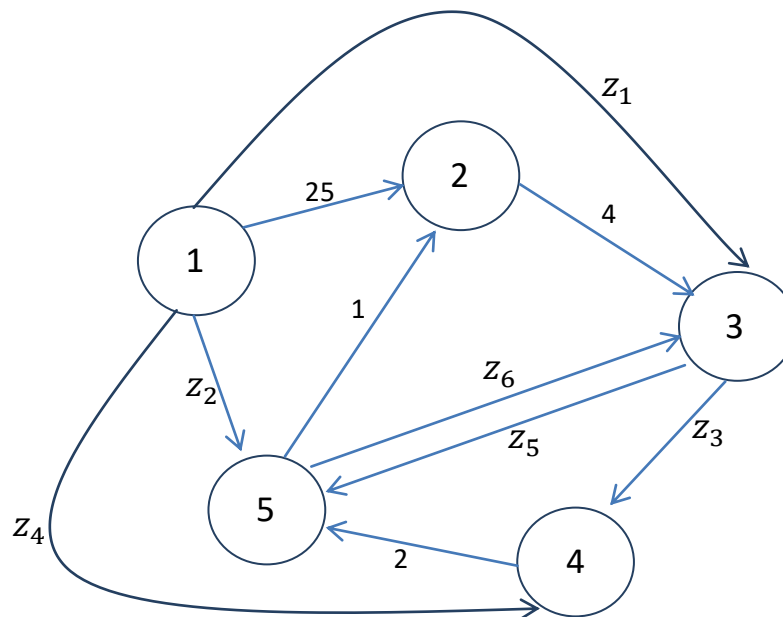
Aufgabe 5 [20]

- [4] Fügen Sie die Werte z_1 bis z_8 aus Aufgabe 1 (in dieser Reihenfolge) in einen zu Beginn leeren binären Suchbaum ein. (Werte können im Baum eventuell mehrfach gespeichert sein.) Skizzieren Sie den Zustand des Baums nach jedem Einfügeschritt.
- [4] Geben Sie in C++ ähnlicher Notation die Definition einer Datenstruktur für einen binären Suchbaum an.
- [8] Geben Sie in C++ ähnlicher Notation eine Definition einer Funktion oder Methode an, die die Höhe des binären Suchbaums ermittelt.
- [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.

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 Dijkstra die jeweils kürzesten Wege vom Knoten 1 zu allen anderen Knoten des Graphen. Wählen Sie eine Notation, aus der die Arbeitsweise des Algorithmus klar ersichtlich wird.
- [4] Ist für den Dijkstra-Algorithmus eher die Verwendung einer Adjazenzmatrix oder einer Adjazenzliste vorteilhaft? Begründen Sie Ihre Aussage.