

Algorithmen und Datenstrukturen (PL.ADS.AD.VO)	schriftliche Einzelpruefung	26.06.2012		1
--	--------------------------------	------------	--	---

23	27	26	17	29	15	12	
+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	+ <input type="text"/>	
							25
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 der zweiten Zeile die Ziffern Ihrer Matrikelnummer in die leeren Kästchen ein. Führen Sie die Additionen durch und ermitteln Sie die Zahlen z_1 bis z_7 . (z_8 ist bereits mit dem fixen Wert 25 belegt.)

Aufgabe 2 [18]

Gegeben sind folgende Funktionen:

```
void g(int n, int i) {
    if (!i) return;
    for (int j=2*n; j>0; --j)
        g(n,i-1);
}

void f(int n) {
    if (!n) return;
    for (int i=0; i<z1%10; i++)
        f(n/(z3%10+2));
    g(n,z2%3);
    f(n/(z3%10+2));
    g(n,z2%3);
}
```

Berechnen Sie die Laufzeit der Funktion f in Θ -Notation abhängig von n. Setzen Sie dazu für z_1 , z_2 und z_3 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) seien in dieser Reihenfolge von links nach rechts in einem Array gespeichert. Sortieren Sie die Werte aufsteigend mit

- [8] Heapsort.
- [8] Quicksort.
- [4] Mergesort.

Geben Sie jeweils alle benötigten Zwischenschritte so genau an, dass der Ablauf des Algorithmus klar ersichtlich wird.

Aufgabe 4 [20]

- [9] Fügen Sie die Werte z_1 bis z_7 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 $g(k) = k\%3 + 1$.

Skizzieren Sie den Zustand der Hashtabelle nach jedem Einfügeschritt.

- [1] Löschen Sie den Wert z_2 aus der Tabelle und skizzieren Sie den Zustand der Hashtabelle.
- [5] Geben Sie den Kollisionspfad (besuchte Indexpositionen) bei einer Suche nach dem Wert z_7 an.
- [5] Geben Sie den Kollisionspfad (besuchte Indexpositionen) bei einer Suche nach dem Wert 42 an.

Aufgabe 5 [20]

- [8] Fügen Sie die Werte z_1 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.
- [4] Geben Sie in C++ ähnlicher Notation die Definition einer Datenstruktur für einen binären Suchbaum an.
- [4] Geben Sie in C++ ähnlicher Notation eine Definition einer Funktion an, die den binären Suchbaum depth first traversiert und alle gespeicherten Werte ausgibt.
- [4] Notieren Sie die Ausgabe Ihrer Funktion, wenn sie für den unter Teilaufgabe a erstellten Baum aufgerufen wird. Handelt es sich dabei um eine preorder Traversierung, eine postorder Traversierung, eine inorder Traversierung oder um eine andere Art der Traversierung?

Algorithmen und Datenstrukturen (PI.ADS.AD.VO)	schriftliche Einzelpruefung	26.06.2012		2
--	--------------------------------	------------	--	---

Aufgabe 6 [20]

Gegeben ist die folgende Adjazenzmatrix mit Wegekosten für einen gerichteten Graphen (die Werte z_1 bis z_8 sind aus Aufgabe 1 zu übernehmen):

$$\begin{pmatrix} 0 & 0 & z_1 & z_2 & z_3 \\ z_4 & 0 & z_5 & z_6 & 0 \\ z_7 & z_8 & 0 & z_1 & z_2 \\ 0 & 0 & z_3 & 0 & z_4 \\ z_5 & 0 & z_6 & z_7 & 0 \end{pmatrix}$$

- [2] Skizzieren Sie den gerichteten Graphen.
- [10] Bestimmen Sie mit dem Algorithmus von Dijkstra die jeweils kürzesten Wege vom Knoten 1 (erste Zeile, erste Spalte der Matrix) zu allen anderen Knoten des Graphen.
- [8] Bestimmen Sie mit dem Algorithmus von Prim 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 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.