Algorithmen und Datenstrukturen (PI.ADS.AD.VO)	schriftliche Einzelpruefung	03.12.2007		1

Aufgabe 1 [20]

a. [15] Gegeben sind folgende Funktionen:

```
void loop(int n,int u) {
  if (u<=0) return;
  for (int i=0; i<n; ++i)
    loop(n,u-1);
}


void f(int n, int geb) {
  if (n<=0) return;

  int u=geb%10+2; //Letzte Stelle von geb plus 2
  int v=geb/10%10+2; //Vorletzte Stelle von geb plus 2
  int w=geb/100%10+2; //Drittletzte Stelle von geb plus 2

  loop(n,u);
  for (int i=0; i<v; ++i)
    f(n/w,geb);
}
```

Bestimmen Sie das Laufzeitverhalten der Funktion f in Theta-Notation abhängig von n, wenn beim Aufruf der Funktion für den Parameter *geb* Ihr Geburtsdatum in der Form JJJJMMTT übergeben wird (Beispiel: Das Geburtsdatum 7. März 1985 ergibt den Wert 19850307 für den Parameter *geb*).

b. [5] Sie sollen für die Bearbeitung einer sehr großen Menge an Datensätzen zwischen zwei möglichen Algorithmen A und B wählen. Sie wissen, dass Algorithmus A eine Laufzeit der Ordnung $O(log(n))$ hat und B eine Laufzeit der Ordnung $O(log(log(n)))$. Für welchen Algorithmus würden Sie sich entscheiden, wenn die Bearbeitung schnellstmöglich durchgeführt werden muss? Könnte es sein, dass sich der von Ihnen gewählte Algorithmus in einem konkreten Testszenario trotzdem als langsamer herausstellt, als der andere? Begründen Sie Ihre Antworten.

Aufgabe 2 [20]

Addieren Sie zu Ihrer Matrikelnummer die Zahl 14278356. Nehmen Sie an, die Ziffern der so erhaltenen Summe wären in einem Feld gespeichert und sollen sortiert werden.

a. [8] Beschreiben Sie, wie das Feld durch den Algorithmus Heapsort sortiert wird. Geben Sie alle benötigten Zwischenschritte so genau an, daß der Ablauf des Algorithmus klar ersichtlich wird.

b. [8] Beschreiben Sie, wie das Feld durch den Algorithmus Mergesort sortiert wird. Geben Sie alle benötigten Zwischenschritte so genau an, daß der Ablauf des Algorithmus klar ersichtlich wird.

c. [4] Was bedeuten die Begriffe "stabil", "instabil", "insitu" und "exsitu" in Bezug auf Sortierverfahren?

Aufgabe 3 [25]

a. [10] Fügen Sie die Ziffern Ihrer Matrikelnummer in der Reihenfolge von links nach rechts in eine zu Beginn leere Hashtabelle der Größe 7 ein. Verwenden Sie die Hashfunktion $h(k) = k \; modulo \; 7$ und linear probing zur Behandlung von Kollisionen. Geben Sie den Zustand der Hashtabelle nach jeder eingefügten Ziffer an.

Was würden Sie an der gegebenen Situation verändern, um die Anzahl der Kollisionen zu verringern?

b. [15] Geben Sie in Java- bzw. C++ - ähnlichem Pseudocode eine Definition für die Datenstruktur einer Hashtabelle an, die als Kollisionsstrategie seperate chaining verwendet. Entwerfen Sie dazu eine passende Methode search, die nach einem bestimmten Schlüsselwert in der Tabelle sucht und true retourniert, falls der Wert gefunden wurde, false sonst.

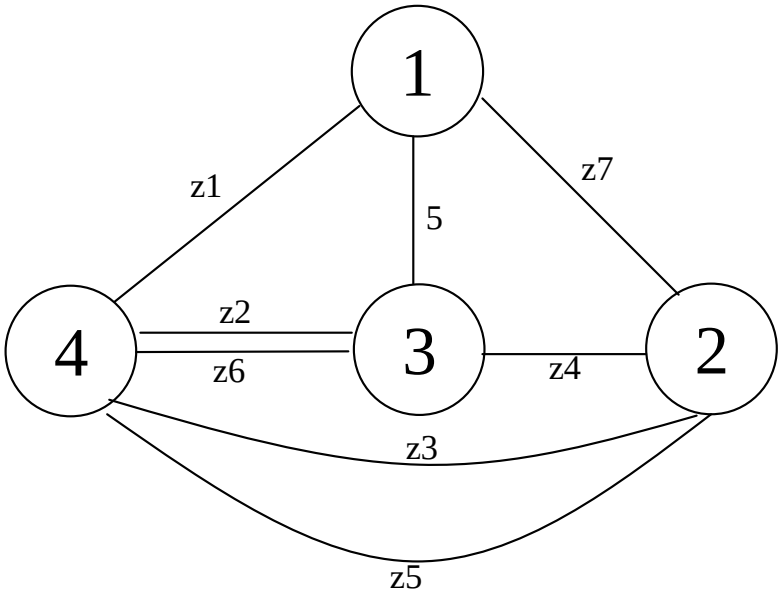
Aufgabe 4 [15]

Gegeben ist der arithmetische Ausdruck $(3 + 8) * 7 - (3 + 5 * 6)/2 + 1$

a. [4] Geben Sie einen Expression Tree für diesen Ausdruck an.

b. [6] Geben Sie die verschiedenen Reihenfolgen an, in denen die Knoten besucht werden, wenn Sie den Baum inorder, preorder und postorder traversieren.

c. [5] Nehmen Sie an, es wäre ein Expression Tree für n binäre Operationen zu erstellen. Welche maximale bzw. minimale Höhe des Expression Trees ist zu erwarten?



Aufgabe 5 [20]

Ersetzen Sie in der Abbildung die Gewichte z1 bis z7 durch Werte, die Sie aus Ihrer Matrikelnummer wie folgt ermitteln: zi ergibt sich aus der i-ten Stelle der Matrikelnummer (von rechts beginnend nummeriert) plus 1. Für die Matrikelnummer 1234567 wäre z2 beispielsweise 7 (=6+1).

a. [15] Ermitteln Sie mit dem Algorithmus von Kruskal einen minmal spannenden Baum für den Graphen.

   Geben Sie die jeweiligen Zwischenergebnisse (Wälder) an, die sich nach dem Hinzufügen jeder einzelnen Kante ergeben. Der Arbeitsablauf des Algorithmus muss aus Ihren Notizen eindeutig nachvollziehbar sein.

b. [5] Geben Sie die beiden Abfolgen von Knoten an, die Sie erhalten, wenn Sie den erhaltenen Baum mit den Methoden breadth-first-search und depth-first-search traversieren. Suchen Sie sich als Startknoten einen Knoten (den gleichen für beide Traversierungen) aus, sodass Ihre beiden Ergebnisse unterschiedlich sind.