Algorithmische Mathematik I

Wintersemester 2009/2010
Prof. Dr. Mario Bebendorf
Dr. Jan Hamaekers

universitätbonn

Nachklausur

16.3.2010

Bearbeitungsdauer: 120 Minuten

Achtung: Bitte beschriften Sie jedes Blatt mit Namen!

Name:	
Vorname:	
Matrikelnr.:	

Note:	

Aufgabe	1	2	3	4	5	6	7	$\sum$
max. Pkt.	10	10	10	10	10	20	10	100
err. Pkt.								

Aufgabe 1. (Gleitpunktdarstellung)

Ein Rechner verwendet 8 Bit zur Darstellung eine normalisierten Gleitkommazahl zur Basis $b = 2$, mit Mantissenlänge $t = 5$ und Exponentenlänge $p = 2$. Hier wird ein Bit v_m für das Vorzeichen der Mantisse, 4 Bit $(m_4\, m_3\, m_2\, m_1)$ für die Mantisse, ein Bit v_e für das Vorzeichen des Exponenten und 2 Bit $(e_1\, e_0)$ für den Exponenten verwendet, d.h.

$$[v_m\, m_4\, m_3\, m_2\, m_1\, v_e\, e_1\, e_0];\ v_m, m_i, v_e, e_j \in \{0,1\}.$$

Beachten Sie, dass das *hidden bit* $m_0 = 1$ feststeht und somit nicht explizit gespeichert wird, sondern es werden nur die vier signifikanten Bits der Mantisse gespeichert.

a) Geben Sie die Formeldarstellung einer solchen Gleitpunktzahl $z \neq 0$ an.

b) Geben Sie den absoluten und den relativen Rundungsfehler bei der Darstellung der Zahlen $z = 3/4$ und $z = 4/3$ an.

c) Geben Sie die größte und die kleinste im Rechner darstellbare positive Gleitpunktzahl $z > 0$ an. Notieren sie jeweils sowohl die Dezimaldarstellung (Sie können auch Brüche angeben) als auch die zugehörige Bitfolge.

d) Definieren Sie den Begriff „Maschinengenauigkeit" und geben Sie die Maschinengenauigkeit $\varepsilon_{\mathbb{F}}$ des Rechners an.

(2+3+3+2=10 Punkte)

Aufgabe 2. (Drei-Term-Rekursion)

Für eine Zahl $x \in \mathbb{R}$ ist die folgende Drei-Term-Rekursion gegeben:

$$p_{n+1} = 2\cos(x)p_n - p_{n-1}$$

mit den Startwerten $p_0 = 1$ und $p_1 = \cos(x)$. Zeigen Sie durch vollständige Induktion, dass $p_n = \cos(nx)$ für alle $n = 0, 1, 2, \ldots$ gilt.
Nutzen Sie dabei das Additionstheorem

$$\cos((n \pm 1)x) = \cos(nx)\cos(x) \mp \sin(nx)\sin(x).$$

(10 Punkte)

Aufgabe 3. (Laufzeitabschätzung)

Ein Algorithmus habe in Abhängigkeit der Eingabegröße n die Laufzeitfunktion $T(n)$. Durch rekursive Aufrufe ergebe sich

$$T(n) \leq \begin{cases} 1, & \text{für } n = 1, \\ n + \max_{1 \leq k < n} \left(T(k) + T(n-k)\right), & \text{für } n > 1. \end{cases}$$

a) Zeigen Sie induktiv, dass $T(n) = O(n^2)$.

b) Geben Sie ohne Beweis die Worst-Case-Komplexität sowie die durchschnittliche Komplexität des Quicksort Algorithmus aus der Vorlesung an. Für welche Mengen tritt der Worst-Case Fall ein ?

(7+3=10 Punkte)

Aufgabe 4. (Cholesky-Zerlegung)

a) Berechnen Sie die Cholesky-Zerlegung $A = LL^T$ der Matrix

$$A = \begin{bmatrix} 4 & 2 & -2 \\ 2 & 2 & -1 \\ -2 & -1 & 17 \end{bmatrix}.$$

b) Lösen Sie mit der Zerlegung aus a) das lineare Gleichungssytem $Ax = b$ mit $b = (14, 20, -7)^T$.

c) Sei $A \in \mathbb{R}^{n \times n}$ eine positiv definite Matrix mit unterer Bandbreite $p \in \mathbb{N}$, d.h. $a_{ij} = 0$ für $i > j + p$. Zeigen Sie, dass der Faktor L ebenfalls untere Bandbreite p hat.

(2+2+6=10 Punkte)

Aufgabe 5. (Programmieraufgabe: Cholesky-Zerlegung)

a) Gegeben sei das C–Fragment

```
void cholesky(double** A, unsigned int n, double** L)
{
}
```

Vervollständigen Sie diese Funktion mit einer Implementierung der Cholesky-Zerlegung einer Eingabematrix $A \in \mathbb{R}^{n \times n}$ und Ausgabematrix $L \in \mathbb{R}^{n \times n}$. Greifen Sie mittels '`A[i][j]`' (bzw. '`L[i][j]`') auf Matrixeinträge zu. Berücksichtigen Sie dabei folgende Punkte:

- Das C–Interface muss eingehalten werden.
- Es soll *keine* Pivotisierung durchgeführt werden.
- Die Einträge oberhalb der Diagonalen von `L` werden als auf 0 gesetzt angenommen.

b) Sei nun

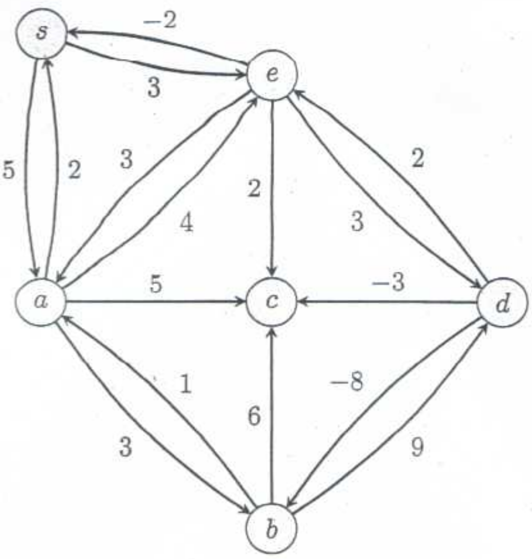
```
void loeseCHLGS(double** L, double* b, unsigned int n)
{
}
```

gegeben. Implementieren Sie ein Verfahren zur Lösung des linearen Gleichungssystems $LL^T x = b$ mithilfe von Vorwärts- und Rückwärtssubstitution, wobei L in der Matrix `L` gegeben ist. Das Eingabeformat ist dasselbe wie das Ausgabeformat aus a). Nach Ende des Verfahrens soll der Vektor `b` mit x überschrieben worden sein. Beachten Sie: auch hier muss das C–Interface eingehalten werden.

(10+10=20 Punkte)

Aufgabe 6. (Graphen)

a) Gegeben sei folgender gewichteter Digraph $G_1 = (V_1, E_1, c_1)$:



Die angegebenen Zahlen geben die Gewichte der jeweiligen Kante an. Berechnen Sie mit einem geeigneten Algorithmus aus der Vorlesung alle kürzesten Wege vom Startkonten s aus. Stellen Sie die Zwischenergebnisse tabellarisch dar.

b) Sei $G = (V, E, c)$ ein beliebiger ungerichteter Graph mit einer positiven Gewichtsfunktion. Angenommen, alle Kantengewichte sind verschieden und es gebe einen Kreis C im Graph G. Zeigen Sie, dass die schwerste Kante e_{max} des Kreises C zu keinem minimalem Spannbaum von G gehört.
Tipp: Es bietet sich ein indirekter Beweis an.

(5+5=10 Punkte)

Aufgabe 7. (Matchings)

Sei $G = (V, E)$ ein ungerichteter Graph. M^* sei ein größtes Matching in G und $x, y \in V$ seien freie Knoten, d.h. Knoten, die von M^* nicht überdeckt[1] werden. Zeigen Sie:

a) Die Knoten x und y sind nicht durch eine Kante verbunden und jeder Knoten $v \in \mathrm{suc}(x) \cup \mathrm{suc}(y)$ wird von M^* überdeckt.

[1] Ein Knoten $v \in V$ wird von einem Matching M überdeckt, falls eine Kante $(u_1, u_2) \in M$ mit $v \in \{u_1, u_2\}$ existiert. Ein Knoten $v \in V$ heißt frei, falls $v \notin e$ für alle $e \in M$.

b) M^* enthält keine Kante, die einen Knoten $u \in \mathrm{suc}(x)$ mit einem Knoten $v \in \mathrm{suc}(y)$ verbindet.

(4+6=10 Punkte)

Viel Erfolg!