Algorithmen und Programmieren 3
erste Nachklausur zur ersten Teilklausur
Donnerstag, 26. Februar 2004.

Anleitung: Streichen Sie *eine* Aufgabe deutlich auf dem Angabeblatt. Diese Aufgabe wird nicht in die Bewertung einbezogen. Wenn Sie selbst keine Aufgabe streichen, wird die erste Aufgabe nicht in die Bewertung einbezogen. Bearbeiten Sie die übrigen drei Aufgaben.

Bearbeiten Sie jede Aufgabe auf einem getrennten Blatt. Jede Aufgabe hat 10 Punkte. Bearbeitungszeit: 90 Minuten

---

1. Wir wollen eine Variante von a-b-Bäumen konstruieren, bei der weniger Speicher verschwendet wird. Es sei $a = 50$. Wie klein kann man b wählen, wenn man

   - beim Einfügen *zwei* Geschwisterknoten zu Hilfe nimmt, falls diese den Überlauf aufnehmen können, und
   - beim Entfernen bis zu *drei* Geschwisterknoten zum Ausborgen in Betracht zieht?

   Begründen Sie Ihre Antwort!

2. Gegeben ist ein Graph mit den Knoten $V = \{1, 2, \ldots, n\}$ und m Kanten in Adjazenzlistenspeicherung. Die Kanten in (i, j) den Adjazenzlisten sollen nach den Endknoten umsortiert werden; das heißt, in der Adjazenzliste von i soll die Kante (i, j) vor (i, k) kommen, wenn $j < k$ ist.

   (a) Skizzieren Sie einen Algorithmus, der diese Sortieraufgabe *in linearer Zeit*, das heißt, in $O(m + n)$ Zeit löst.

   (b) Schreiben Sie ein Java-Programm für diese Aufgabe.

3. Eliminieren Sie die Endrekursion aus dem folgenden Programmstück, das eine verkettete Liste umkehrt. Gehen Sie dabei systematisch Schritt für Schritt vor, und geben Sie alle Zwischenschritte bei der Transformation explizit an. Schreiben Sie am Ende eine einzige Methode als Ersatz für `umkehre`.

```
class Liste { Object wert; Liste next; }

static Liste umkehre(Liste x)
{ if(x==null) return x;
  else return umkehre3(null, x, x.next);
}
static Liste umkehre3(Liste davor, Liste x, Liste danach)
{ x.next = davor;
  if (danach==null) return x;
  else return umkehre3(x, danach, danach.next);
}
```

4. Schreiben Sie ein Java-Programm, das die Knoten eines binären Baumes in *Niveau-Reihenfolge* ausgibt, das heißt, zuerst die Wurzel, dann die Knoten mit Tiefe 1, dann mit Tiefe 2, und so weiter.

   Ihr Programm soll lineare Laufzeit haben.