Prof. Dr. R. Schrader
Birgit Engels [engels@zpr.uni-koeln.de]

WS 2006/2007

Theoretische Informatik

5. Übung, Abgabe Mittwoch, 22.11.2006

Aktuelle Informationen bezüglich der Vorlesung und der Übungen finden sich unter :
http://www.zaik.uni-koeln.de/AFS/teachings/courses/ThInf/index.html und
http://www.zaik.uni-koeln.de/AFS/teachings/courses/ThInf/uebungen.html

Aufgabe 18 (NP-Vollständigkeit — 3-Occurence-SAT — 6 Punkte):

a) Zeigen Sie (durch Reduktion auf **SAT**), dass folgendes Problem NP-schwer ist:

   L-3-Occurence-SAT: Gegeben sei eine boolsche Formel $\alpha = \bigwedge c_i, c_i = \bigvee l_{ij}$ in konjunktiver Normalform, wobei jedes *Literal* l_{ij} (Es gilt $l_{ij} = x$ und $l_{kl} = \neg x$ werden separat gezählt!) höchstens 3 mal in ganz α auftritt. Ist α erfüllbar?

b) Betrachten Sie folgende Abwandlung des obigen Problems:

   V-3-Occurence-SAT: Gegeben sei eine boolsche Formel $\alpha = \bigwedge c_i, c_i = \bigvee l_{ij}$ in konjunktiver Normalform, wobei jede *Variable* l_{ij} (Es gilt $l_{ij} = x$ und $l_{kl} = \neg x$ werden gemeinsam gezählt!) höchstens 3 mal in ganz α auftritt. Ist α erfüllbar?

   Ist die Reduktion aus a) für das höchstens 3-malige Auftreten einer *Variable* in α übertragbar? Erweitern bzw. verändern Sie Ihre Beweisidee, um **V-3-Occurence-SAT** $\in NP$ zu zeigen!

Hinweis: Beachten Sie, dass auch Erweiterungen von α in KNF ausgedrückt werden müssen (siehe Aufgabe 18 :)!

Aufgabe 19 (NP-Vollständigkeit — Konjunktive und Disjunktive Normalform boolscher Ausdrücke — 6 Punkte):
Nach dem *Satz von Cook* (siehe Vorlesung) ist das (KNF-)**SAT**-Problem, also für eine gegebene boolsche Formel α_{KNF} in konjunktiver Normalform ($\alpha_{KNF} = \bigwedge c_i, c_i = \bigvee l_{ij}$) zu entscheiden, ob eine erfüllende Belegung existiert, NP-vollständig. Zu jeder solchen KNF α_{KNF} gibt es eine äquivalente Formel in disjunktiver Normalform $\alpha_{DNF} = \bigvee c_i, c_i = \bigwedge l_{ij}$, die genau dann erfüllbar ist, wenn α_{KNF} erfüllbar ist.

a) Ist das Erfüllbarkeitsproblem für boolsche Formeln in DNF (**DNF-SAT** analog zu **SAT**) ebenfalls NP-vollständig ? (Geben Sie einen Beweis für die NP-Vollständigkeit von **DNF-SAT** *oder* einen polynomiellen Lösungsalgorithmus für das Problem an !)

Bitte wenden!

b) Ist/Wäre "**DNF-SAT** $\in P$" ein Widerspruch zum *Satz von Cook* ? (Begründen Sie Ihre Antwort.)

Aufgabe 21 (Probabilistische Algorithmen — Randomisiertes Sortieren — 4 Punkte):
Gegeben sei folgendes randomisiertes Sortierverfahren (RandQS—Randomisiertes Quicksort): Die n zu sortierenden Zahlen seien im Feld S[1:n] gespeichert.

- Wähle zufällig einen Index $1 \leq k \leq n : S[k]$ ist Pivotelement.
- Verwalte Zeiger i, der S von links nach rechts durchläuft bis ein $S[i]$ mit $S[i] > S[k]$ gefunden ist.
- Verwalte Zeiger j, der S von rechts nach links durchläuft bis ein $S[j]$ mit $S[j] \leq S[k]$ gefunden ist.
- Falls $j > i$ werden $S[j]$ und $S[i]$ vertauscht und beide Indizes durchlaufen S weiter.
- Falls $j < i$ werden $S[k]$ und $S[i-1]$ vertauscht.
- Wähle für $S[1:k]$ und $S[k+1:n]$ neue Pivotelemente und sortiere die Teilfelder rekursiv.

a) Welcher probabilistischen Komplexitätsklasse gehört **RandQS** an (Begründung)?

b) Analysieren Sie die *worst case* Laufzeit von **RandQS**!
Hinweis: Zeigen Sie mittels Induktion, dass

$$T(n) = \frac{(n+1)(n+2)}{2} - 3$$

eine obere Schranke für die Anzahl der Vergleichsoperationen für die Sortierung eines Feldes $S[1:n]$ ist.

Aufgabe 22 (Probabilistische Algorithmen — Hashing — 4 Punkte):
Wir betrachten ein einfaches Hashing-Verfahren mit Hash-Funktion $h(x) = x \bmod m$, welches die Schlüssel x nach $h(x)$ in m Listen einsortiert (Kollisionsbehandlung durch verkettete Listen). Weiterhin betrachten wir eine Folge von n Schlüsseln x_i, $1 \leq i \leq n$, wobei das Einsortieren von x_i in alle Listen l_j, $1 \leq j \leq m$, gleich wahrscheinlich sein soll.

a) Wie groß ist die Wahrscheinlichkeit, dass Liste l_j leer bleibt ?

b) Wie groß sind die Wahrscheinlichkeiten für mindestens eine Kollision und genau eine Kollision in Liste l_j ?