Prof. Dr. G. Schnitger
M. Poloczek

Frankfurt, den 24.02.2011

Klausur
Algorithmentheorie

WS 2010/2011

Name: _____ Vorname: _____

Matrikelnummer: _____ Geburtsdatum: _____

Studiengang: _____

⇓ BITTE GENAU LESEN ⇓

Die Klausur besteht aus **4** Aufgaben, in denen maximal **100 Punkte** erreicht werden können. Die Klausur ist mit Sicherheit bestanden, wenn zusammen mit der Bonifikation aus den Übungen mindestens **50 Punkte** erreicht werden.

Bitte schreibe oben auf **jede** Seite in Blockschrift Namen und Matrikelnummer. Überprüfe, ob die Klausur aus insgesamt **11** durchnummerierten Seiten besteht. Bitte benutze die Rückseiten. Weitere Blätter sind ggf. erhältlich.

Schreibe **nicht** mit Bleistift. Ein DIN A4 Blatt mit Notizen ist das einzige zugelassene Hilfsmittel.

Eine umgangssprachliche, aber **strukturierte** Beschreibung von Algorithmen ist völlig ausreichend.

Werden zu einer Aufgabe zwei Lösungen angegeben, so gilt die Aufgabe als nicht gelöst. Entscheide Dich also immer für **eine** Lösung.

Die Klausur dauert 180 Minuten. Der Termin für die Einsichtnahme ist Mittwoch, der 02. März von 10:00-11:00 Uhr im SR 307.

! VIEL ERFOLG !

1a	1b	1c	1d	2a	2b	2c	3a	3b	4a	4b	4c	Σ
5	6	6	8	6	6	13	10	15	4	6	15	100

bestanden	nicht bestanden

Bonifikation	Σ

Name:	Matrikelnummer:

AUFGABE 1 — Sortieren

(a, 5 Punkte) In der Vorlesung haben wir Quicksort nicht-rekursiv mit Hilfe eines Stacks implementiert. Welches Teilproblem, das größere oder das kleinere, sollten wir auf den Stack legen, um die Stackhöhe möglichst gering zu halten? Welche Schranke kannst Du dann für die maximale Stackhöhe angeben? Hier sind Antworten ohne Begründung ausreichend.

(b, 6 Punkte) In einem Array sind n ganze Zahlen gespeichert. Beschreibe einen möglichst effizienten Algorithmus, der die Zahlen so umsortiert, dass die negativen vor den positiven stehen. Bestimme die Laufzeit Deines Algorithmus.

(c, 6 Punkte) In einem Array sind n ganze Zahlen gespeichert. Nun wollen wir eine Zahl bestimmen, die am häufigsten auftritt. Entwirf einen möglichst effizienten Algorithmus für diese Fragestellung. Bestimme die Laufzeit Deines Algorithmus.

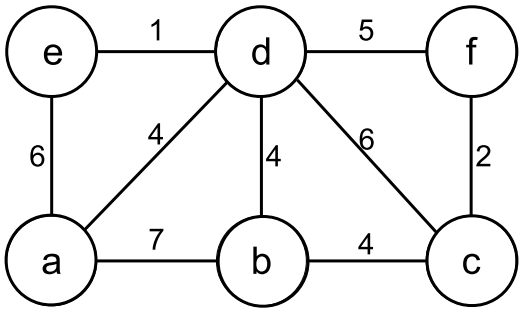
Name:	Matrikelnummer:

(d, 8 Punkte) Um nachzuweisen, dass jedes vergleichsorientierte Sortierverfahren mindestens $\Omega(n \log n)$ Vergleiche für das Sortieren von n Schlüsseln benötigt, haben wir **Sortierbäume** betrachtet. Gib eine informelle Definition für Sortierbäume an und zeige, dass jeder Sortierbaum für n Schlüssel mindestens die Tiefe $\Omega(n \log n)$ besitzt.

Name:	Matrikelnummer:

AUFGABE 2 — Graphalgorithmen

Gegeben sei der folgende Graph.



(a, 6 Punkte) Gib an, in welcher Reihenfolge die Knoten von Prims Algorithmus ausgewählt werden. Startknoten ist der Knoten **a**. Gib den resultierenden minimalen Spannbaum an.

(b, 6 Punkte) Gib an, in welcher Reihenfolge die Knoten vom Algorithmus von Dijkstra abgearbeitet werden, wenn **a** zum Startknoten gemacht wird. Gib den resultierenden Baum der kürzesten Wege an.

Name:	Matrikelnummer:

(c, 13 Punkte) Theo Rhetiker hat eine Reihe von kleinen karibischen Inseln erworben und wohnt auf Insel s. Die Inseln sind durch eine Vielzahl von waghalsigen, unterschiedlich langen Brückenkonstruktionen verbunden. Da Theo es zwar liebt, mit seinem Pickup spazieren zu fahren, ihm jedoch lange Fahrten über Wasser unangenehm sind, möchten wir für jede Insel u einen Weg berechnen, so dass **die Länge der längsten Brücke** entlang des Weges vom Wohnort s nach u minimal ist.

1.) Modelliere die Fragestellung als Graphproblem und entwirf einen möglichst effizienten Algorithmus dafür.

2.) Zeige die Korrektheit Deines Algorithmus und analysiere seine Laufzeit.

Name:	Matrikelnummer:

AUFGABE 3 — Entwurfsmethoden

(a, 10 Punkte) Wir erhalten einen gerichteten Graphen G mit Knotenmenge $V = \{1, \ldots, n\}$; zusätzlich sei länge(u, v) die nichtnegative Länge der Kante von u nach v.

Floyds Algorithmus berechnet die Länge eines kürzesten Wegen von Knoten u nach Knoten v für alle Knoten $u, v \in V$ in Zeit $O(n^3)$.

- Beschreibe die **Teilprobleme**, die Floyds Algorithmus löst und gib jeweils die **Rekursionsgleichungen** an.
- Warum ist die Laufzeit $O(n^3)$?

Name:	Matrikelnummer:

(b, 15 Punkte) Wir starten in einer Position $(1, i)$ auf der ersten Zeile und möchten eine Position (n, j) auf der letzten Zeile in einem $n \times n$ Gitter G_n erreichen. G_n besteht aus den n^2 Knoten (i, j) mit $1 \leq i, j \leq n$. Wenn wir uns in Position (i, j) befinden, dann können wir in einem Schritt

- Position $(i + 1, j)$ erreichen (falls $i < n$) und erhalten $p_{i,j}$ Euro,
- oder Position $(i + 1, j - 1)$ erreichen (falls $i < n$ und $j > 1$) und erhalten $q_{i,j}$ Euro
- oder Position $(i + 1, j + 1)$ erreichen (falls $i < n$ und $j < n$) und erhalten $r_{i,j}$ Euro.

Entwirf einen möglichst effizienten Algorithmus, der einen wertvollsten Weg von einem Knoten der ersten Zeile zu einem Knoten der letzten Zeile bestimmt. Gib einen Korrektheitsbeweis und analysiere die Laufzeit.

Name:	Matrikelnummer:

AUFGABE 4 — $\mathcal{NP}$– Vollständigkeit

(a, 4 Punkte) Definiere die polynomielle Reduktion:

$K \leq_p L \Leftrightarrow$

(b, 6 Punkte) Angenommen, die Sprache L ist $\mathcal{NP}$-vollständig.

Zeige: Wenn L einen effizienten Algorithmus besitzt, wenn also $L \in \mathcal{P}$ gilt, dann folgt $\mathcal{P} = \mathcal{NP}$.

Name:	Matrikelnummer:

(c, 15 Punkte) In dem Problem **Set Packing** erhalten wir endliche Mengen $A_1, \ldots, A_m$ sowie eine Zahl k. Die Eingabe $(A_1, \ldots, A_m, k)$ ist genau dann zu akzeptieren, wenn es k paarweise disjunkte Mengen $A_{i_1}, \ldots, A_{i_k}$ gibt: Für $r \neq s$ muss also $A_{i_r} \cap A_{i_s} = \emptyset$ gelten.

Zeige die Reduktion **Independent Set** $\leq_{\mathbf{p}}$ **Set Packing**. Begründe deine Antwort.

(Zur Erinnerung: Im Independent Set Problem ist ein ungerichteter Graph (V, E) gegeben sowie eine Zahl l. Die Eingabe (G, l) ist genau dann zu akzeptieren, wenn es eine Menge von l Knoten gibt, so dass keine zwei Knoten dieser Menge durch eine Kante miteinander verbunden sind.)

Name:	Matrikelnummer:

Name:	Matrikelnummer: