Prof. Dr. G. Schnitger
M. Poloczek

Frankfurt, den 07.04.2011

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.

Bitte beschreibe stets kurz Deinen Lösungsansatz. 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 20. April von 9:00-10:00 Uhr im SR 307.

! VIEL ERFOLG !

1a	1b	1c	2a	2b	2c	3a	3b	4a	4b	4c	Σ
6	9	10	6	6	13	10	15	4	6	15	100

bestanden	nicht bestanden

Bonifikation	Σ

Name:	Matrikelnummer:

AUFGABE 1 — Sortieren

a.) **Beschreibe** einen möglichst schnellen Algorithmus, der den k-kleinsten Schlüssel aus einer nicht-sortierten Folge von n Zahlen bestimmt. (Eine Laufzeitanalyse ist nicht verlangt.)

b.) Wir erhalten k aufsteigend sortierte Folgen mit je n Zahlen. Entwirf einen möglichst schnellen Algorithmus, der die k Folgen zu einer einzigen, aufsteigend sortierten Folge *mischt*.

Beschreibe deinen Algorithmus und **analysiere** seine Laufzeit.

Name:	Matrikelnummer:

c.) Das Aufrufprotokoll einer populären Webseite erfasst von jedem Besucher den Zeitpunkt des Aufrufs der Seite und den Moment, zu dem der Besucher die Seite verlässt. Sei n die Zahl der Besucher im Laufe eines Tages, seien (a_i, w_i) die Ankunfts- bzw. die Weggehzeiten des $i$ten Besuchers.

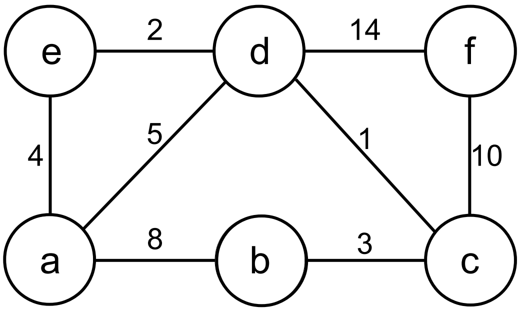
Für eine statistische Auswertung soll die maximale Zahl M von Besuchern, die gleichzeitig auf der Seite waren, bestimmt werden.

Beschreibe einen Algorithmus, der die maximale Besucherzahl M in Zeit $O(n \cdot \log(n))$ bestimmt. Benutze dabei Pseudocode, beschreibe kurz die Idee Deines Algorithmus und begründe, warum die gestellte Aufgabe richtig gelöst und die Zeitschranke eingehalten wird.

Name:	Matrikelnummer:

AUFGABE 2 — Graphalgorithmen

Gegeben sei der folgende Graph.



a.) **Gib an**, in welcher Reihenfolge die Kanten von Kruskals Algorithmus ausgewählt werden. Gib den resultierenden minimalen Spannbaum an.

b.) **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.) Gegeben sei ein Rechnernetzwerk $N = (V, E)$, das aus einer Menge V von Rechnern und einer Menge E von Verbindungen zwischen den Rechnern besteht. Jeder Verbindung $\{u, v\}$ von Rechnern u und v ist eine Zuverlässigkeit $z_{u,v}$ mit $0 \leq z_{u,v} \leq 1$ zugeordnet: Wir interpretieren $z_{u,v}$ als die Wahrscheinlichkeit, dass die Verbindung $\{u, v\}$ *nicht* versagt. Die Zuverlässigkeit eines Weges definieren wir als das Produkt der Zuverlässigkeiten der Kanten des Weges.

Entwirf einen möglichst effizienten Algorithmus, der Wege größter Zuverlässigkeit von einem ausgezeichneten Rechner $s \in V$ zu allen anderen Rechnern bestimmt. **Bestimme** die Laufzeit Deines Verfahrens für n Rechner und m Verbindungen.

Name:	Matrikelnummer:

AUFGABE 3 — Entwurfsmethoden

a.) Die *natürlichen* Zahlen $x_1, \dots, x_n$ sowie die natürliche Zahl B sind gegeben. Es ist zu entscheiden, ob es eine Teilmenge $S \subseteq \{1, \dots, n\}$ mit $\boxed{\sum_{i \in S} x_i = B}$ gibt.

Beschreibe einen Algorithmus, der dieses Problem in Zeit $O(n \cdot B)$ löst. Welche Teilprobleme löst Dein Algorithmus und welche Rekursionsgleichung benutzt Du?

Name:	Matrikelnummer:

b.) Theo besitzt einige karibischen Inseln $I_1, \dots, I_n$. Einige dieser Inseln sind durch Brücken verbunden: Wenn die Inseln I_a und I_b durch eine Brücke verbunden sind, dann bezeichnet Länge$(a, b) =$ Länge(b, a) die Länge dieser Brücke. Die Scheu vor langen Brücken hat Theo zwar abgelegt, trotzdem möchte er nicht mehr als k Brücken überqueren, um von seiner Hauptinsel s zur jeweiligen Zielinsel zu gelangen.

Entwirf einen möglichst schnellen dynamischen Programmieralgorithmus, der für *jede* Insel I_j einen Weg mit den folgenden Eigenschaften bestimmt:

(1) Der Weg führt von der Hauptinsel s zur Insel I_j und überquert höchstens k Brücken.

(2) Der Weg ist ein kürzester Weg unter allen Wegen mit Eigenschaft (1).

Zeige die Korrektheit Deines Verfahrens und **bestimme** seine Laufzeit.

Name:	Matrikelnummer:

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

a.) **Zeige** oder **widerlege**: Alle Sprachen in $\mathcal{P}$ gehören auch zur Klasse $\mathcal{NP}$.

b.) **Zeige**: Wenn K eine $\mathcal{NP}$-vollständige Sprache ist und wenn $K \leq_P L$ gilt, dann ist L eine $\mathcal{NP}$-harte Sprache.

Transitivität kann angenommen werden: Wenn $L_1 \leq_P L_2$ und $L_2 \leq_P L_3$ für Sprachen L_1, L_2 und L_3 gilt, dann gilt auch $L_1 \leq_p L_3$.

Name:	Matrikelnummer:

c.) Im Problem **VC** ist ein Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$ gegeben. Es ist zu entscheiden, ob es eine Menge $W \subseteq V$ von k Knoten gibt, so dass alle Kanten $e \in E$ mindestens einen Endpunkt in W besitzen. In der Vorlesung wurde gezeigt, dass VC ein $\mathcal{NP}$-vollständiges Problem ist.

Im Problem **Hitting Set** sind ein Universum U, Teilmengen $S_1, \ldots, S_m \subseteq U$ und eine Zahl $k \in \mathbb{N}$ gegeben. Es ist zu entscheiden, ob es eine Teilmenge $T \subseteq U$ mit $|T| \leq k$ gibt, so dass T jede Teilmenge S_i in mindestens einem Element schneidet.

Zeige die Reduktion **VC** $\leq_P$ **Hitting Set**.

Name:	Matrikelnummer: