Prof. Dr. G. Schnitger Frankfurt, den 13.02.2009

Klausur
Algorithmentheorie

WS 2008/2009

Name: ______________________ Vorname: ______________________

Matrikelnummer: ______________________

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 schreiben Sie oben auf **jede** Seite in Blockschrift Namen und Matrikelnummer. Überprüfen Sie, ob die Klausur aus insgesamt **11** durchnummerierten Seiten besteht.

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

Am Ende der Klausur befinden sich noch zwei leere Seiten, benutzen Sie bitte auch Rückseiten. Weitere Blätter sind ggf. erhältlich.

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

Werden zu einer Aufgabe 2 Lösungen angegeben, so gilt die Aufgabe als nicht gelöst. Entscheiden Sie sich also immer für **eine** Lösung.

Die Klausur dauert 180 Minuten. Der Termin für die Einsichtnahme ist Donnerstag, der 19. Februar von 10-11:30 Uhr im SR 307.

! VIEL ERFOLG !

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

bestanden	nicht bestanden

Bonifikation	Σ

Name:	Matrikelnummer:

AUFGABE 1 **Sortieren**

(a, 4 Punkte) In der Vorlesung haben wir Quicksort nicht-rekursiv mit Hilfe eines Stacks implementiert. Wir haben erreicht, dass der Stack höchstens logarithmisch groß wird.

Welches Teilproblem, das größere oder das kleinere, wird auf den Stack gelegt?

Das größere Teilproblem ○ das kleinere Teilproblem ○

(b, 6 Punkte) Welche der Sortierverfahren **Quicksort**, **Bubblesort** und **Mergesort** sind in-place Verfahren, benutzen also höchstens $n + o(n)$ Speicherplatz, um n Zahlen zu sortieren?

Quicksort: ○ ja ○ nein, Bubblesort: ○ ja ○ nein, Mergesort: ○ ja ○ nein,

(c, 5 + 5 Punkte)] Gegeben sind zwei Arrays A und B mit insgesamt n Schlüsseln. Wir möchten feststellen, ob $A \subseteq B$ gilt, d.h. ob alle Elemente aus A auch in B enthalten sind.

(1) Entwerfen Sie einen möglichst schnellen Algorithmus und analysieren Sie seine Laufzeit.

(2) Wir mehmen jetzt an, dass alle Schlüssel zur Menge $\{1, \ldots, n^5\}$ gehören. Entwerfen Sie einen möglichst schnellen Algorithmus für die neue Situation und analysieren Sie seine Laufzeit.

Name:	Matrikelnummer:

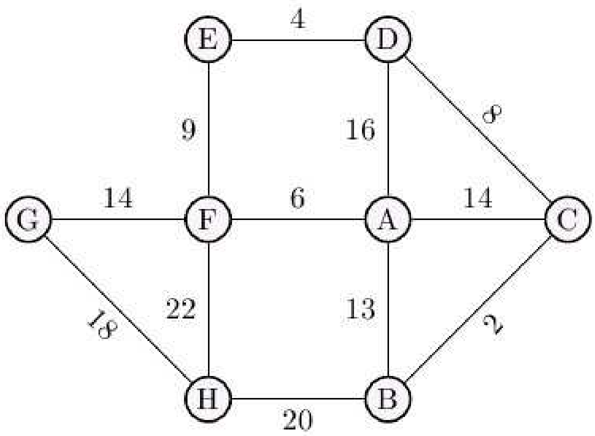
(d, 5 Punkte) Um nachzuweisen, dass jedes vergleichsorientierte Sortierverfahren mindestens $\Omega(n \cdot \log_2 n)$ Vergleiche für das Sortieren von n Schlüsseln benötigt, haben wir Sortierbäume betrachtet.

Beweisen Sie, dass jeder Sortierbaum für n Schlüssel mindestens die Tiefe $\Omega(n \cdot \log_2 n)$ besitzt.

Name:	Matrikelnummer:

AUFGABE 2 Graphalgorithmen

(a, 5+5 Punkte) Gegeben sei der folgende Graph.



(1) Geben Sie an, in welcher Reihenfolge die Knoten von Prims Algorithmus ausgewählt werden. Startknoten ist der Knoten A. Geben Sie den resultierenden minimalen Spannbaum an.

(2) Geben Sie an, in welcher Reihenfolge die Knoten vom Algorithmus von Dijkstra abgearbeitet werden, wenn A zum Startknoten gemacht wird. Geben Sie den resultierenden Baum der kürzesten Wege an.

Name:	Matrikelnummer:

(b, 15 Punkte) Der ungerichtete Graph $G = (V, E)$ modelliert ein Rechnernetz: Die Knoten in V entsprechen also den Rechnern und die Kanten in E entsprechen den Direktverbindungen zwischen Rechnern.

Jeder Kante $e \in E$ ist eine Bandbreite $b(e) \geq 0$ zugewiesen. Wir definieren die Bandbreite eines Weges W als die kleinste Bandbreite einer Kante in W.

(1) Ein Knoten s ist ausgezeichnet. **Entwerfen** Sie einen möglichst schnellen Algorithmus, der für jeden Knoten $v \in V$ die größte Bandbreite eines Weges von s nach v bestimmt.

(2) **Begründen** Sie die Korrektheit ihrer Lösung und **analysieren** Sie die Laufzeit.

Name:	Matrikelnummer:

AUFGABE 3 — Entwurfsmethoden

(a, 10 Punkte) In einer Gallerie sollen Gemälde bedeutender Künstler ausgestellt werden. Es versteht sich von selbst, dass die Versicherung die Auflage macht, dass jedes Werk im Blickfeld mindestens eines Wachmanns liegt. Da Personal teuer ist, werden Sie gebeten, die Anzahl notwendiger Wachleute zu mininieren.

Als Ausstellungsraum wurde ein langer, gerader Gang gewählt. Die Positionen der Gemälde sind als aufsteigende Folge $0 \leq x_1 \leq x_2 \leq \cdots \leq x_n$ reeller Zahlen gegeben. Der Standort eines Wachmanns wird ebenfalls durch eine reelle Zahl $y \geq$ festgelegt: Ein Wachmann in Position y garantiert die Sicherheit der Gemälde im Intervall $[y-1, y+1]$.

Beschreiben Sie einen möglichst schnellen Algorithmus, welcher die minimale Anzahl benötigter Wachleute bestimmt, **geben** Sie die Laufzeit an und **begründen** Sie seine Korrektheit.

Name:	Matrikelnummer:

(b, 15 Punkte) Sie führen ein Team von IT-Beratern und müssen für jede der nächsten n Wochen festlegen, welcher Job zu erledigen ist. Dabei haben Sie für Woche i die Wahl zwischen einem konventionellen Job, der einen Gewinn von $l_i > 0$ Euro bringt, und einem komplexen Job mit einem Gewinn von $h_i > 0$ Euro ($1 \leq i \leq n$). Der Haken ist allerdings, dass das Team für komplexe Jobs eine Woche Vorbereitungszeit benötigt: Wird also in Woche i ein komplexer Job angenommen, dann darf in Woche $i-1$ kein Job angenommen werden.

Für Woche 1 darf kein komplexer Job angesetzt werden. Andererseits ist es stets in Ordnung, einen konventionellen Job für Woche i anzusetzen, egal welcher Art von Tätigkeit das Team in der vorausgehenden Woche nachgegangen ist.

Beschreiben Sie einen effizienten Algorithmus, der für Eingaben $l_1, l_2, \ldots, l_n$ und $h_1, h_2, \ldots, h_n$ den *Wert* eines Arbeitsplans mit höchstem Gewinn bestimmt. **Bestimmen** Sie die Laufzeit ihrer Lösung.

Name:	Matrikelnummer:

AUFGABE 4 $\mathcal{NP}$-Vollständigkeit

(a, 10 Punkte) Beantworten Sie die folgenden Fragen mit Wahr oder Falsch. Eine Begründung ist **nicht** erforderlich. Es kann angenommen werden, dass $\mathcal{P} \neq \mathcal{NP}$ gilt.

Für jede richtige Antwort werden 2 Punkte vergeben. Eine falsche Antwort erhält -2 Punkte. Keine Antwort ergibt 0 Punkte. Die Mindestpunktzahl für diese Aufgabe ist 0 Punkte.

- Wenn L_1 und L_2 in $\mathcal{P}$ liegen, dann ist auch $L_1 \cap L_2$ in $\mathcal{P}$.
  ○ Wahr ○ Falsch
- Wenn L_1 und L_2 in $\mathcal{NP}$ liegen, dann ist auch $L_1 \cap L_2$ in $\mathcal{NP}$.
  ○ Wahr ○ Falsch
- Wenn L_1 und L_2 $\mathcal{NP}$-vollständig sind, dann ist auch $L_1 \cap L_2$ $\mathcal{NP}$-vollständig.
  ○ Wahr ○ Falsch
- Wenn $L_1 \leq_P L_2$ und $L_2 \in \mathcal{NP}$, dann auch $L_1 \in \mathcal{NP}$.
  ○ Wahr ○ Falsch
- Die Sprache $L = \{G \mid G \text{ ist 2-färbbar}\}$ aller ungerichteten und 2-färbbaren Graphen liegt in $\mathcal{P}$. (Ein Graph ist k-färbbar, wenn die Knoten so mit k Farben markiert werden können, dass keine Kante gleichgefärbte Endpunkte besitzt.)
  ○ Wahr ○ Falsch

Name:	Matrikelnummer:

(b, 3+12 Punkte) Im Problem der binären Programmierung sind eine reellwertige $m \times n$ Matrix A sowie ein reellwertiger Vektor b der Länge m gegeben. Gesucht wird ein Vektor

$$z \in \{0,1\}^n,$$

der das Ungleichungssystem $Az \geq b$ löst, insbesondere müssen also alle Ungleichungen

$$\sum_{j=1}^{n} A[i,j] \cdot z_j \geq b_i$$

für $i = 1, \ldots, m$ erfüllt werden.

(1) Zeigen Sie, dass das Problem der binären Programmierung in $\mathcal{NP}$ liegt.

(2) Weisen Sie nach, dass das Problem der binären Programmierung $\mathcal{NP}$–hart ist. Für eine Reduktion bietet sich beispielsweise 3-SAT an.

Name:	Matrikelnummer:

Name:	Matrikelnummer: