Name:............................. **Matr.-Nr.:** ....
Studiengang:......... .... **Angestr. Abschl.:** ....

	Punkte	max
1		6
2		6
3		6
4		6
5		6
6		6
Σ		60

Probeklausur Algorithmen

(Prof. Kaufmann/Czarnetzki/Güler, Sommersemester 2016)

- Die Bearbeitungszeit beträgt 90 Minuten, elektronische Hilfmittel sind verboten, nur ein handbeschriebenes A4-Blatt mit Notizen ist erlaubt.
- Von den 6 Aufgaben mit jeweils 6 möglichen Punkten werden lediglich die 5 besten bewertet.

Viel Erfolg!

Aufgabe 1: **(Dijkstra)** (6 Punkte)

Gegeben Sei ein gerichteter Graph $G = (V, E, w)$, wobei $w\colon E \to \mathbb{N}$ eine Kantengewichtsfunktion ist. Beweisen oder widerlegen Sie folgende Aussagen:

(a) Sei $c \in \mathbb{N}_{>0}$ und $w_a(e) = w(e) + c$ für alle $e \in E$. Dijkstra findet auf $G_a = (V, E, w_a)$ die gleichen kürzesten Wege wie auf G.

(b) Sei $c \in \mathbb{N}_{>0}$ und $w_b(e) = c \cdot w(e)$ für alle $e \in E$. Dijkstra findet auf $G_a = (V, E, w_a)$ die gleichen kürzesten Wege wie auf G.

Aufgabe 2: **(Rekursion)** (6 Punkte)

Sei $n = (3/2)^k$ mit $k \in \mathbb{N}$. Folgende Rekursion ist für die Funktion T gegeben:

$T(1) = 1$

$T(n) = 2 \cdot T(2/3n)$.

Bestimmen Sie den Funktionswert von $T(n)$ möglichst genau in Abhängigkeit von n. Beweisen Sie die Korrektheit, ohne das Mastertheorem zu benutzen.

Aufgabe 3: **(Greedy-Algorithmen)** (9 Punkte)

Sei $P = (\{1, \dots, n\}, \{\{i, i+1\} |\text{ für } i = 1, \dots, n-1\}, w)$ ein Graph mit **Knotengewichten**, das heißt $w\colon V \to \mathbb{N}$ ordnet jedem Knoten ein Gewicht zu. Ein Maximum Weight Independent Set von P ist eine Menge von paarweise nicht benachbarten Knoten mit maximalem Gewicht.

Hinweis: Machen Sie sich zunächst klar, um welche Art von Graphen es sich bei P handelt.

(a) Betrachten Sie folgende Greedy-Algorithmen für das Problem

i. $S \leftarrow \emptyset$
while $V \neq \emptyset$ **do**
  $v \leftarrow \max_{v \in V}(w(v))$
  $S \leftarrow S \cup \{v\}$
  Lösche v und seine Nachbarn aus V
end while
return S

ii. $S_0 \leftarrow \{i |\ i \text{ ist gerade}\}$
$S_1 \leftarrow \{i |\ i \text{ ist ungerade}\}$
if $\sum_{v \in S_0} w(v) > \sum_{v \in S_1} w(v)$ **then**
  return S_0
else
  return S_1
end if

Zeigen Sie, dass beide Algorithmen nicht immer ein Maximum Weight Independent Set von P finden.

(b) Geben Sie mittels dynamischer Programmierung einen Algorithmus an, der das Problem korrekt löst.

Aufgabe 4: **(Suchbaumeigenschaft)**

Gegeben sei ein Binärbaum, der die natürlichen Zahlen $S = \{1, 2, \ldots, 1000\}$ enthält. Angenommen wir wollen die Zahl 363 in dem Suchbaum finden. Welche der folgenden Sequenzen von Knoten könnte bei der Suche nach 363 **nicht** auftreten?

(a) 924, 220, 911, 244, 898, 258, 362, 363

(b) 925, 202, 911, 240, 912, 245, 363

(c) 935, 278, 347, 621, 299, 392, 358, 363

Begründen Sie Ihre Antowort!

Aufgabe 5: **(Algorithmenentwurf - 3-Zykel)**

Geben Sie einen Algorithmus in Pseudocode an, der zu einem ungerichteten Graphen $G = (V, E)$ alle 3-Zykel in G ausgibt.