

Prof. Petra Mutzel
Gunnar Klau
Ivana Ljubić
René Weisskircher

Sommersemester 2000/2001

**Prüfung zur Vorlesung
Algorithmen und Datenstrukturen 2
9. März 2001**

1. Machen Sie bitte die folgenden Angaben in deutlicher Blockschrift:

Name: _____ Vorname: _____

Matrikelnummer: _____ Studienkennzahl: _____

2. Legen Sie während der Vorlesungsprüfung Ihren Studentenausweis vor sich auf das Pult.
3. Schreiben Sie die Lösungen direkt auf das jeweilige Aufgabenblatt. Wenn Ihnen das Papier ausgeht, bitten Sie die Aufsicht um Nachschub. Es ist nicht erlaubt, eigenes Papier zu verwenden!
4. Denken Sie daran, dass keinerlei Hilfsmittel erlaubt sind (weder Taschenrechner, irgendwelche Unterlagen, Handys,...).

VOR DER ABGABE AUSZUFÜLLEN:

5. Geben Sie bitte die Anzahl der zusätzlich abgegebenen Blätter an: _____
6. Kreuzen Sie bitte die von Ihnen bearbeiteten Aufgaben in der ersten Zeile der Tabelle an:

Aufgabe	A 1	A 2	A 3	A 4	A 5	A 6	Σ	Note
bearbeitet							—	—
maximale Punktzahl	10	6	6	15	10	13	60	—
erreichte Punktzahl								

Viel Erfolg!

Aufgabe 1: Depth-First Search**10 Punkte**

Gegeben sei ein ungerichteter Graph $G = (V, E)$. Verändern Sie den Pseudo-Code für den DFS-Algorithmus, um jeden Knoten $v \in V$ mit einem Label $c(v) \in 1 \dots k$ zu markieren, so dass

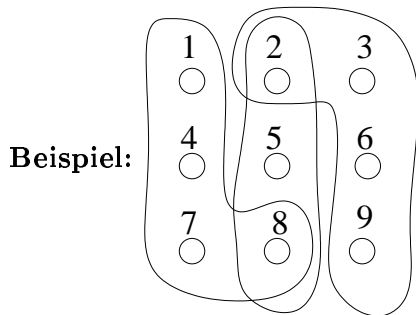
$$c(u) = c(v) \Leftrightarrow u \text{ und } v \text{ sind in derselben zusammenhängenden Komponente.}$$

k ist die Anzahl von zusammenhängenden Komponenten in G .

Aufgabe 2: Branch & Bound (Set-Covering Problem)**6 Punkte**

Gegeben sei eine endliche Menge X , und eine Familie \mathcal{F} von Teilmengen von X so dass jedes Element von X zu mindestens einer Teilmenge in \mathcal{F} gehört. Das Problem ist eine Teilmenge $\mathcal{C} \subset \mathcal{F}$ zu finden, die möglichst wenige Teilmengen von X enthält, und X überdeckt ("covers"), d. h. :

$$X = \bigcup_{S \in \mathcal{C}} S.$$



$$X = \{1, 2, \dots, 9\},$$

$$\mathcal{F} = \{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}, \{9\}, \{2, 3, 6, 9\}, \{2, 5, 8\}, \{1, 4, 7, 8\}\}.$$

$$\text{Lösung: } \mathcal{C} = \{\{2, 3, 6, 9\}, \{2, 5, 8\}, \{1, 4, 7, 8\}\}.$$

$$|\mathcal{C}| = 3.$$

Sie lösen das Set-Covering Problem mittels Branch-and-Bound. An jedem Knoten können Sie folgende Schranken berechnen:

☐ globale untere Schranke☐ globale obere Schranke☐ lokale untere Schranke☐ lokale obere und globale untere Schranke

(Bitte kreuzen Sie richtige Antworten an!)

Aufgabe 3: WhatAmI**6 Punkte**

Gegeben sei folgender Algorithmus für den gewichteten Graphen $G = (V, E)$ mit den Gewichten $l : E \mapsto \mathbb{R}^+$.

input: Graph $G = (V, E)$. Zwei Knoten $s, t \in V$.
output: Kürzester Weg in G zwischen Knoten s und t .
ALGORITHM(G, s, t)
 $T = \{s\}$.
 $dist[s] = 0, dist[v] = l(s, v), \forall (s, v) \in E$, und $dist[v] = \infty$, sonst.
while $t \notin T$ **do**
 Finde $m \in V \setminus T$ s. d. $dist[m] = \min\{dist[v] \mid v \in V \setminus T\}$;
 $T = T \cup \{m\}$;
 for jede Kante $(m, v) \in E$ **do**
 if $v \notin T$ **then**
 if $dist[v] > dist[m] + l(m, v)$ **then**
 $dist[v] = dist[m] + l(m, v)$;
 end if
 end if
 end for
end while
return $dist[t]$;

Es handelt sich hierbei um:

☐ einen Dynamischen Programmierungsalgorithmus.

☐ einen exakten Algorithmus.

☐ eine Verbesserungsheuristik.

☐ ein Austauschverfahren.

(Bitte kreuzen Sie richtige Antworten an!)

Aufgabe 4: Dynamische Programmierung**15 Punkte**

Wenden Sie den Dynamische Programmierungsalgorithmus auf das folgende Beispiel für das 0/1 Rucksackproblem an (Grenzgewicht $K = 38$). Veranschaulichen Sie **alle** Schritte, die zur Konstruktion einer Lösung führen.

Gewichte	8	16	21	17	12
Werte	8	14	16	11	7

Abbildung 1: Rucksackproblem - Beispiel

Aufgabe 5: Suchen in Texten**10 Punkte**

Sie möchten in dem Text:

ACACTAACACATACGTAA

das Muster

ACGA

suchen. Dazu verwenden Sie den Algorithmus von Boyer–Moore. Zeichnen Sie die verschiedenen Verschiebepositionen des Musters ein. Tragen Sie an jeder Verschiebepositionen die Anzahl der Text–Muster–Zeichen–Vergleiche ein. Wie viele sind es insgesamt?

Ein Text–Muster–Zeichen–Vergleich ist ein Vergleich zwischen je einem Zeichen im Text und einem Zeichen im Muster.

Aufgabe 6: Scan-line Prinzip (Sichtbarkeitsproblem)**13 Punkte**

Zwei Liniensegmente s und s' in einer gegebenen Menge horizontaler Liniensegmente S sind gegenseitig *sichtbar*, wenn es eine vertikale Gerade gibt, die s und s' , aber kein weiteres Liniensegment der Menge S zwischen s und s' schneidet.

- a.) Geben Sie möglichst effizienten Pseudocode an, der alle Paare von gegenseitig sichtbaren Liniensegmenten berechnet.

input: Menge $S = \{s_1, \dots, s_n\}$ von horizontalen Liniensegmenten in der Ebene, die durch die Anfangs- und Endpunkte repräsentiert sind;

output: Menge aller Paare von gegenseitig sichtbaren Elementen in S .

SCAN-LINE-PRINZIP(S)

...

- b.) Geben Sie die Laufzeit Ihres Algorithmus für den Worst-Case-Fall in Θ -Notation an.