

Prof. Dr. Sándor P. Fekete
Dr. Christian Scheffer

Klausur
Algorithmen und Datenstrukturen II
10. August 2015

Name: *Mit der Veröffentlichung des Klausurergebnisses nur mit der Matrikelnummer über die Mailingliste und auf der Homepage bin ich einverstanden.*

Vorname:

Matr.-Nr.:

Studiengang: *Unterschrift*

Bachelor Master Diplom Andere

Hinweise:

- Bitte das Deckblatt vollständig ausfüllen.
- Die Klausur besteht aus 13 Blättern, bitte auf Vollständigkeit überprüfen.
- Erlaubte Hilfsmittel: Keine.
- Eigenes Papier ist nicht erlaubt.
- Die Rückseiten der Blätter dürfen beschrieben werden.
- Die Heftung der Blätter darf nicht entfernt werden.
- Die Klausur ist mit 50% der Punkte bestanden.
- Antworten die *nicht* gewertet werden sollen bitte deutlich durchstreichen.
- Mit *Bleistift* oder in *Rot* geschriebene Klausurteile können nicht gewertet werden.
- Werden mehrere Antworten gegeben, werten wir die mit der geringsten Punktzahl.
- Die Bearbeitungszeit für die Klausur ist 120 Minuten.

Aufgabe	1	2	3	4	5	Σ
Punkte	30	20	15	25	10	100
Erzielte Punkte						

Aufgabe 1: Knapsack**(5+5+10+10 Punkte)**

Gegeben ist folgende BINARY-KNAPSACK-Instanz mit 6 Gegenständen:

i	1	2	3	4	5	6
Wert p_i	3	1	2	5	7	3
Gewicht z_i	2	3	3	5	4	4

Dabei darf der Rucksack mit einem Gesamtgewicht von maximal $Z = 14$ bepackt werden.

- a) Ermittle mithilfe des Greedy-Algorithmus (GREEDY_0) aus der Vorlesung eine zulässige Lösung für die gegebene Instanz. Gib in jeder Iteration den aktuellen Gegenstand an, sowie die Menge der bereits gepackten Gegenstände mitsamt ihrem Gesamtgewicht und -wert.

- b) Interpretiere die oben gegebene Instanz als Instanz des FRACTIONAL-KNAPSACK-Problems und berechne mithilfe des Greedy-Algorithmus aus der Vorlesung eine zulässige Lösung. Gib dabei in jeder Iteration den aktuellen Gegenstand, zu welchen Teilen er gepackt wird, sowie den Gesamtzustand an. (Was ist gepackt und wie hoch ist der aktuelle Gesamtwert und -gewicht?) Ist diese Lösung optimal? Begründe Deine Aussage.

c) Wende den Dynamic-Programming-Algorithmus aus der Vorlesung auf die gegebene Instanz an.

Gib die dafür nötige Tabelle komplett mit allen Einträgen an und bearbeite die Gegenstände in der Reihenfolge ihrer Indizes.

d) Betrachte jetzt Branch-and-Bound.

- (i) Welche untere Schranke lässt sich aus a) ableiten? Für welche Teilbäume gilt sie?
- (ii) Welche obere Schranke lässt sich aus b) ableiten? Für welche Teilbäume gilt sie?
- (iii) Welche obere Schranke lässt sich für den Teilbaum mit $x_1 = 1$, $x_2 = 1$, $x_3 = 1$ und $x_4 = 1$ angeben? Was lässt sich daraus im Hinblick auf die aus den vorigen Teilaufgaben bekannten Schranken folgern?
- (iv) Welche untere Schranke lässt sich für den Teilbaum mit $x_1 = x_3 = x_4 = x_5 = 1$ und $x_2 = x_6 = 0$ angeben? Was lässt sich daraus im Hinblick auf die aus den vorigen Teilaufgaben bekannten Schranken folgern?

Aufgabe 2: Approximationsalgorithmen

(15+5 Punkte)

a) Führe Algorithmus 1.26 (GREEDY_k) mit $k = 2$ für die Eingabe $n = 5$, $z_1 = 1$, $z_2 = 1$, $z_3 = 2$, $z_4 = 1$, $z_5 = 1$, $p_1 = 3$, $p_2 = 2$, $p_3 = 3$, $p_4 = 1$, $p_5 = 5$ und $Z = 3$ aus. Gib für jede Iteration die folgenden Mengen bzw. Werte an:

- \bar{S} ,
- $\sum_{i \in \bar{S}} z_i$,
- $Z - \sum_{i \in \bar{S}} z_i$,
- $A_{\bar{S}} := \text{GREEDY}_0(\{z_i | i \notin \bar{S}\}, Z - \sum_{i \in \bar{S}} z_i, \{p_i | i \notin \bar{S}\})$
- $\sum_{i \in \bar{S}} p_i + A_{\bar{S}}$,
- G_k und
- S .

- b) Inwiefern ähnelt Algorithmus GREEDY_k für $k := n$ dem aus der Vorlesung bekannten Verfahren Branch-and-Bound? Inwiefern unterscheiden sich beide Verfahren? Begründe Deine Antwort.

Aufgabe 3: Standortprobleme**(5+10 Punkte)**

Der Median m einer Menge $\{x_1, \dots, x_n\}$ erfüllt definitionsgemäß

$$|\{x_i \mid x_i < m\}| \leq \frac{n}{2} \quad \text{und} \quad |\{x_i \mid x_i > m\}| \leq \frac{n}{2}.$$

- a) Betrachte die Menge $M = \{17, 21, 71, 5, 10, 38\}$. Gib die Menge aller Mediane von M an!

- b) Was passiert mit der Laufzeit des Algorithmus „Median der Mediane“, wenn man jeweils Dreier- statt Fünfergruppen verwendet? Begründe Deine Antwort.

Was passiert mit der Laufzeit, wenn man Siebener- statt Fünfergruppen verwendet? Begründe Deine Antwort.

Aufgabe 4: Dynamische Programmierung (3+3+3+3+3+3+2+5 Punkte)

Betrachte folgendes Problem: Gegeben ist eine Menge $\mathcal{M} := \{[b_i, e_i] \mid 1 \leq i \leq n\}$ von n Intervallen. Für $i \in \{1, \dots, n\}$ sei $w_i > 0$ das Gewicht des Intervalles $[b_i, e_i]$. Gesucht ist eine Teilmenge $\mathcal{M}' \subseteq \mathcal{M}$, so dass die Intervalle aus \mathcal{M}' sich nicht schneiden und so dass die Summe der Gewichte der Intervalle aus \mathcal{M}' maximal ist.

Zur Vereinfachung nehmen wir an, dass die Intervalle aufsteigend bzgl. e_i nummeriert sind. D.h. es gilt $e_1 \leq e_2 \leq \dots \leq e_n$.

Betrachte für das obige Problem das folgende dynamische Programm:

$$OPT(i) = \begin{cases} \max\{OPT(i-1), w_i + OPT(pred(i))\} & \text{falls } i > 0 \\ 0 & \text{falls } i = 0 \end{cases},$$

wobei $pred: \{1, \dots, n\} \rightarrow \{0, \dots, n-1\}$, wie folgt definiert sei

$$i \mapsto \begin{cases} j & \text{falls } j = \max_{k \in \{1, \dots, n\}} \{k \mid e_k \leq b_i\} \text{ definiert ist} \\ 0 & \text{sonst} \end{cases}.$$

- a) Gib die Werte $pred(1), \dots, pred(6)$ für folgendes Beispiel an: $[b_1 = 1, 3 = e_1]$, $[b_2 = 2, 6 = e_2]$, $[b_3 = 3, 7 = e_3]$, $[b_4 = 5, 9 = e_4]$, $[b_5 = 8, 11 = e_5]$ und $[b_6 = 10, 12 = e_6]$.

- b) Beschreibe in eigenen Worten, was allgemein $pred(i)$ für eine Bedeutung für das Intervall $[b_i, e_i]$ hat.

c) Gib einen Algorithmus an, der $pred$ berechnet.

d) Gib die Werte $OPT(0), \dots, OPT(6)$ für das Beispiel aus Aufgabenteil a) mit den Gewichten $w_1 = 2, w_2 = 3, w_3 = 2, w_4 = 4, w_5 = 3$ und $w_6 = 3$ an.

e) Beschreibe in eigenen Worten, welche Bedeutung $OPT(i)$ für $i \in \{0, \dots, n\}$ hat.

f) Gib einen Algorithmus an, der unter Verwendung der bereits berechneten Werte $pred[0], \dots, pred[n]$ das obige dynamische Programm realisiert, also alle Werte $OPT(0), \dots, OPT(n)$ berechnet.

g) Welche Laufzeit hat Dein Algorithmus?

h) Beweise mittels vollständiger Induktion über n , dass $OPT(n)$ der Wert einer optimalen Lösung ist.

Aufgabe 5: Komplexität

(2+2+2+2+2 Punkte)

- a) Was bedeutet es, wenn ein Problem zur Klasse NP gehört?
- b) Was bedeutet es, wenn ein Problem NP-vollständig ist?
- c) Wie beweist man, dass ein Problem NP-vollständig ist?
- d) Was ist der Unterschied zwischen polynomiell und pseudopolynomiell?
- e) Wir wollen eine Instanz von INTEGER KNAPSACK optimal lösen. Nenne je eine Situation, in der
- Branch-and-Bound einen Vorteil hat
 - Dynamic Programming einen Vorteil hat.

Viel Erfolg 😊