Algorithmen und Datenstrukturen II — **SoSe 2020**

Prof. Dr. Sándor Fekete — **Abgabe:** 21.05.20

Arne Schmidt — **Rückgabe:** ab 27.05.20

Übungsblatt 1

Abgabe der Lösungen muss bis zum 21.05.20 um 23:59 Uhr erfolgen. Lösungen müssen per Mail mit einer pdf-Datei (Name der Datei „`blatt_[nr]_[grp]_[team].pdf`“) an den jeweiligen Tutor geschickt werden. Email-Adressen sind unter `https://www.ibr.cs.tu-bs.de/alg/index.html` zu finden.

Hausaufgabe 1 (Fractional Knapsack): **(5 Punkte)**

In dieser Aufgabe betrachten wir Fractional Knapsack. Sei $Z = 49$, und seien die sieben Objekte mit folgenden Werten gegeben:

i	1	2	3	4	5	6	7
z_i	10	15	13	7	10	14	12
p_i	6	10	13	10	8	10	9

Wende den Greedy-Algorithmus für Fractional Knapsack aus der Vorlesung auf diese Instanz an. Gib in jeder Iteration den aktuellen Gegenstand, den Anteil x_i zu dem er gepackt wird, sowie den Gesamtzustand (aktueller Gesamtwert und aktuelles Gesamtgewicht) an.

Hausaufgabe 2 (Maximum Knapsack): **(7+5 Punkte)**

In dieser Aufgabe betrachten wir Maximum Knapsack. Um dieses Problem zu lösen, wandeln wir den Algorithmus für Fractional Knapsack aus der Vorlesung folgendermaßen ab: Wir sortieren die Objekte aufsteigend nach $\frac{z_i}{p_i}$. Dann gehen wir die Objekte der Sortierung entsprechend durch und packen ein Objekt in den Rucksack, falls es hineinpasst. Eine formale Beschreibung ist in Algorithmus 1 zu sehen. Wir nennen diesen Algorithmus Greedy$_0$.

function Greedy$_0(z_1, \ldots, z_n, Z, p_1, \ldots, p_n)$
  Sortiere Objekte nach $\frac{z_i}{p_i}$ aufsteigend; dies ergibt Permutation $\pi(1), \ldots, \pi(n)$.
  for $j := 1$ **to** n **do**
    if $\left(\sum_{i=1}^{j-1} x_{\pi(i)} z_{\pi(i)} + z_{\pi(j)} \leq Z\right)$ **then**
      $x_{\pi(j)} := 1$
    else
      $x_{\pi(j)} := 0$
  return $x_1, \ldots, x_n$

Algorithmus 1: Greedy$_0$

a) Sei $Z = 28$, und seien die fünf Objekte mit folgenden Werten gegeben:

i	1	2	3	4	5
z_i	21	14	8	7	6
p_i	21	15	9	6	2

Wende $\textsc{Greedy}_0$ auf diese Instanz an. Entscheide in jeder Iteration, ob der aktuelle Gegenstand gepackt wird und gib den Gesamtzustand (aktueller Gesamtwert und aktuelles Gesamtgewicht) an. Ist die erhaltene Lösung optimal? Begründe deine Antwort!

b) Zeige: Die ganzzahlige Lösung P_G von $\textsc{Greedy}_0$ kann beliebig weit von einer optimalen Lösung P_{OPT} entfernt sein, d.h.: Für jedes $0 < c \leq 1$ gibt es eine Instanz, sodass $0 < \frac{P_G}{P_{OPT}} < c$.
(Hinweis: Es wird vorausgesetzt, dass $z_i < Z$ für alle $1 \leq i \leq n$ gilt.)

Hausaufgabe 3 (Hörsaal-Belegung): (3+4 Punkte)
Betrachte das Hörsaal-Belegungsproblem (siehe große Übung #1):

Gegeben: Menge von Intervallen $\mathcal{I} := \{I_1 = [s_1, e_1), \ldots, I_n = [s_n, e_n)\}$

Gesucht: Eine größtmögliche Teilmenge $\mathcal{I}' \subseteq \mathcal{I}$ von disjunkten Intervallen.

a) Zeige: Das Problem kann nicht optimal gelöst werden, wenn sukzessive das kürzeste, noch nicht überdeckte Intervall aufgenommen wird.

b) Zeige: Die Strategie aus a) liefert eine Lösung $\mathcal{I}'_{\text{ALG}}$, die mindestens halb so groß ist wie eine optimale Lösung $\mathcal{I}'_{\text{OPT}}$, also $|\mathcal{I}'_{\text{ALG}}| \geq \frac{1}{2}|\mathcal{I}'_{\text{OPT}}|$
(Hinweis: Wie viele Intervalle der optimalen Lösung kann ein Intervall aus $\mathcal{I}'_{\text{ALG}}$ schneiden?)

Hausaufgabe 4 (Dynamische Programmierung für $\textsc{Subset Sum}$): (6 Punkte)
Wende das dynamische Programm für $\textsc{Subset Sum}$ auf folgende Instanz an:

$Z = 15$ und

i	1	2	3	4	5
z_i	4	6	2	4	1

Fülle hierzu die Tabelle 1 aus, wobei der Eintrag in Zeile i und Spalte x dem Wert $\mathcal{S}(x, i)$ entspricht. Nullen müssen nicht eingetragen werden.

i \ x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
1																
2																
3																
4																
5																

Tabelle 1: Tabelle für das dynamische Programm für $\textsc{Subset Sum}$.