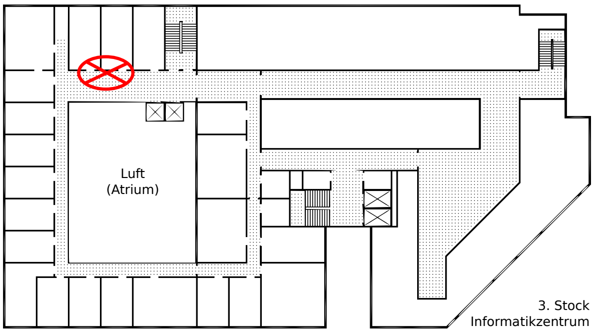
Abteilung Algorithmik **Sommer 2018**
Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig

Prof. Dr. Sándor P. Fekete
Arne Schmidt

Algorithmen und Datenstrukturen II
Übung 1 vom 19. 04. 2018

Abgabe der Lösungen bis zum Donnerstag, den 03. 05. 2018 um 13:15 im Hausaufgabenschrank bei Raum IZ 337. Es werden nur mit einem dokumentenechten Stift (kein Rot!) geschriebene Lösungen gewertet.
Bitte die Blätter zusammenheften und vorne deutlich mit eigenem Namen, Matrikel- und Gruppennummer, sowie Studiengang versehen!



Dieses Blatt ist in drei Bereiche unterteilt: Allgemeiner Teil (A), theoretischer Teil (T) und praktischer Teil (P). Abgegeben muss der A-Teil und entweder T oder P. Mögliche Kombinationen sind also A und T bzw. A und P mit einem Gesamtwert von jeweils 30 Punkten. Die Auswahl gilt nur für dieses Blatt, d.h. auf dem nächsten Blatt kann sich wieder umentschieden werden.

A — Allgemeiner Teil

Aufgabe 1 (Fractional Knapsack): **(8+4 Punkte)**
In dieser Aufgabe betrachten wir Fractional Knapsack.

a) Sei $Z = 100$, und seien die Objekte $1, \dots, 7$ folgendermaßen gewählt:

i	1	2	3	4	5	6	7
p_i	10	13	8	12	15	5	5
z_i	24	26	20	20	20	20	16

Führe Algorithmus 1.4 (Greedy für Fractional Knapsack) auf diese Instanz aus. Gib in jeder Iteration den aktuellen Gegenstand, zu welchen Teilen er gepackt wird, sowie den Gesamtzustand (was ist gepackt, wie hoch ist der aktuelle Gesamtwert und -gewicht) an.

b) Sei P_I die Lösung von Algorithmus 1.4 für eine Knapsack-Instanz I. Zeige: $\lfloor P_\mathrm{I} \rfloor$ ist eine obere Schranke für Maximum Knapsack.

Aufgabe 2 (Maximum Knapsack): **(7+3 Punkte)**
In dieser Aufgabe betrachten wir Maximum Knapsack. Um dieses Problem zu lösen, wandeln wir den Algorithmus 1.4 (Greedy für Fractional Knapsack) folgendermaßen um: 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 $\textsc{Greedy}_0$.

a) Sei $Z = 23$, und seien die Objekte $1, \dots, 6$ folgendermaßen gewählt:

i	1	2	3	4	5	6
p_i	2	9	16	4	9	1
z_i	4	8	23	6	5	3

Führe $\textsc{Greedy}_0$ auf diese Instanz aus. Gib in jeder Iteration den aktuellen Gegenstand, zu welchen Teilen er gepackt wird, sowie den Gesamtzustand (was ist gepackt, wie hoch ist der aktuelle Gesamtwert und -gewicht) 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. $0 < \dfrac{P_G}{P_{OPT}} \leq \varepsilon$ für beliebiges $0 < \varepsilon \leq 1$.
(Hinweis: Es wird vorausgesetzt, dass mindestens ein Element gepackt wird.)

function $\textsc{Greedy}_0(z_1, \dots, z_n, Z, p_1, \dots, p_n)$
    Sortiere Objekte nach $\frac{z_i}{p_i}$ aufsteigend; dies ergibt Permutation $\pi(1), \dots, \pi(n)$.
    Setze $j := 1$
    while $(j \leq n)$ **do**
        if $\left(\sum_{i=1}^{j-1} x_{\pi(i)} z_{\pi(i)} + z_{\pi(j)} \leq Z\right)$ **then**
            $x_{\pi(j)} := 1$
        end if
        $j := j + 1$
    end while
end function

Algorithmus 1: $\textsc{Greedy}_0$

T — Theoretischer Teil

Aufgabe 3 (Partition): **(3+5 Punkte)**
In dieser Aufgabe betrachten wir das Problem Partition etwas genauer.

a) Angenommen, wir haben einen Algorithmus $\mathcal{A}$, der 0-1-Knapsack löst. Wie können wir $\mathcal{A}$ benutzen, um das Problem Partition zu lösen?

b) Betrachten wir nun einen Brute-Force-Algorithmus für Partition, d.h. einen Algorithmus, der jede mögliche nicht-äquivalente Aufteilung der Elemente durchprobiert. Zwei Aufteilungen sind äquivalent, wenn die Elemente in identische Teilmengen aufgeteilt sind. Zum Beispiel ist eine Aufteilung von 1, 2, 3, 4 in $S_1 = \{1, 2\}$ und $S_2 = \{3, 4\}$ äquivalent zur Aufteilung in $\tilde{S}_1 = \{3, 4\}$ und $\tilde{S}_2 = \{1, 2\}$.

(i) Gib alle nicht-äquivalenten Aufteilungen der folgenden drei Objekte an. Existiert für diese Instanz von PARTITION eine Lösung? Begründe Deine Antwort!

i	1	2	3
z_i	2	3	3

(ii) Zeige: Es gibt 2^{n-1} nicht-äquivalente Aufteilungen

P — Praktischer Teil

Aufgabe 4 (Implementierung KNAPSACK): (8 Punkte)
Implementiere sowohl den fraktionalen Greedy aus der Vorlesung als auch den GREEDY_0 aus Aufgabe 2. Für die fraktionale Variante soll der Anteil des zuletzt gewählten Objekts auf mindestens zwei Stellen genau berechnet werden.

Nutze dazu die Javavorlage und die Testfälle, die auf der Vorlesungsseite[1] zur Verfügung stehen.

a) Löse **alle** gegebenen Instanzen (die Lösungen werden in csv-Dateien gespeichert).

b) Wie gut ist die Lösung der ganzzahligen Variante (GREEDY_0) gegenüber der fraktionalen Variante? Erstelle dazu folgenden *Scatter-Plot*[2]:

- Die x-Achse entspricht der Anzahl der Objekte
- Die y-Achse entspricht dem Quotienten aus dem Wert des ganzzahligen Greedy und dem fraktionalen Greedy.

Wir testen Deine Software mit
javac -cp *:. Knapsack1234567.java && java *:. Knapsack1234567 instance_xyz

Zur Abgabe: Ersetze 1234567 durch Deine Matrikelnummer und gib sowohl die Javadatei, die vom Programm generierten csv-Dateien als auch den zu konstruierenden Scatter-Plot (z.B. als pdf-Datei) per Mail an Deinen entsprechenden Betreuer ab. Nenne in der Mail Name, Matrikel- und Gruppennummer. Es gilt dieselbe Frist wie für die anderen Aufgaben.

(*Hinweis:* Der Wert der Lösung für die fraktionale Variante ist bereits abgerundet!)

[1] http://www.ibr.cs.tu-bs.de/courses/ss18/aud2/
[2] Es wird eine Übung zu Verarbeiten von Daten am 18.04.18 geben.