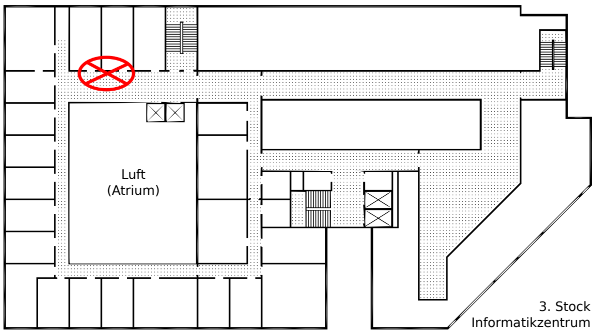
Abteilung Algorithmik **Sommer 2015**
Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig

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

Algorithmen und Datenstrukturen II
Übung 1 vom 22. 04. 2015

Abgabe der Lösungen bis zum Montag, den 04. 05. 2015 um 14:00 im Hausaufgabenrückgabeschrank.



Bitte die Blätter vorne deutlich mit eigenem Namen sowie Matrikel- und Gruppennummer versehen!

Aufgabe 1 (Knapsack): Betrachte einen Algorithmus A, der eine beliebige Probleminstanz P_A (der Größe n) von Problem 1.2' als Eingabe bekommt und für P_A eine optimale Lösung in $T_A(n)$ Zeit berechnet. Wie kann A verwendet werden um, eine Probleminstanz von Problem 1.7 zu lösen? Welche Laufzeit hat Dein Algorithmus? Begründe Deine Antworten. **(4 Punkte)**

Aufgabe 2 (Fractional Knapsack): In dieser Aufgabe geht es um Algorithmus 1.4 (Greedy-Algorithmus für Fractional Knapsack) aus der Vorlesung.

a) Beweise oder widerlege: Eine Lösung für das Problem Fractional Knapsack ist immer eindeutig.

b) Führe Algorithmus 1.4 für die folgende Eingabe aus:

   (a) $n = 10$

   (b) $Z = 50$

   (c) $Z_1 = 10, Z_2 = 14, Z_3 = 15, Z_4 = 1, Z_5 = 21,$
   $Z_6 = 8, Z_7 = 41, Z_8 = 19, Z_9 = 20, Z_{10} = 1$

   (d) $P_1 = 29, P_2 = 2, P_3 = 31, P_4 = 15, P_5 = 51,$
   $P_6 = 19, P_7 = 25, P_8 = 11, P_9 = 7, P_{10} = 1$

c) Welche Laufzeit hat Algorithmus 1.4? Begründe Deine Antwort.

(2+3+1 Punkte)

Aufgabe 3 (Kürzeste gewichtete Pfade): Ein gerichteter Graph $G = (V, E)$ heißt zusammenhängend, falls für alle Knotenpaare $v_1, v_2 \in V$ eine Kantenfolge von v_1 nach v_2 und eine Kantenfolge von v_2 nach v_1 existiert. Es sei ein zusammenhängender, gewichteter, gerichteter Graph $G = (V, E, w)$ gegeben: $V = \{v_1, ..., v_n\}$ ist die Knotenmenge, $E \subseteq V \times V$ die Kantenmenge und $w : E \to \mathbb{R}$ eine Funktion, die jede Kante auf ein reelles Gewicht abbildet. Die gewichtete Länge einer Kantenfolge $P = (e_1, ..., e_k)$ in G (der Länge k) ist die Summe $\sum_{i=1}^{n} w(e_1)$ der Gewichte der Kanten in P. Der Graph G enthalte keine Kantenfolge, die ein Kreis ist und deren gewichtete Länge negativ ist.

Es sei ein Startknoten $s \in V$ gegeben, von dem aus kürzeste, gewichtete Kantenfolgen zu allen anderen Knoten berechnet werden sollen.

a) Beweise: Falls G einen Kreis mit negativer, gewichteter Länge hat, so ist für alle $v \in V \setminus \{s\}$ die gewichtete Länge einer kürzesten, gewichteten Kantenfolge zwischen s und v nicht definiert.

b) Für $i \in \mathbb{N}$ und einen Knoten $v \in V$ sei $v(i)$ die gewichtete Länge einer kürzesten, gewichteten Kantenfolge wischen s und v, die aus maximal i Kanten besteht. Wie kann basierend auf $v_1(i), ..., v_n(i)$ der Wert $v_j(i + 1)$ für $i \in \mathbb{N}$ und $j \in \{1, ..., n\}$ in linearer Zeit berechnet werden. Begründe Deine Antwort.

c) Beweise: Da G keinen Kreis mit negativer, gewichteter Länge hat, existiert für alle $v \in V \setminus \{s\}$ eine kürzeste, gewichtete Kantenfolge, die keinen Kreis enthält.

d) Gib, basierend auf b) und c), einen effizienten Dynamic-Programming-Algorithmus an, der für alle $v \in V \setminus \{s\}$ die Länge einer kürzesten, gewichteten Kantenfolge zwischen s und v berechnet. Warum arbeitet Dein Algorithmus korrekt? Welche Laufzeit hat Dein Algorithmus? Begründe Deinen Antworten.

(2+3+2+3 Punkte)