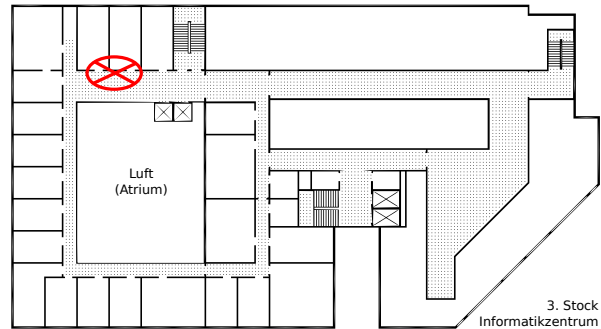


Prof. Dr. Sándor P. Fekete
Dr. Christian Scheffer

Algorithmen und Datenstrukturen II Übung 3 vom 25.05.2016

Abgabe der Lösungen bis zum Montag,
den 06.06.2016 um 13:15 im Hausaufga-
benrückgabeschrank.

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



Aufgabe 1 (Dynamic Programming für Knapsack): Betrachte Algorithmus 1.15 (DP für Knapsack) aus dem Skript.

a) Wende Algorithmus 1.15 auf folgende Knapsackinstanz mit $Z = 13$ an:

i	1	2	3	4	5	6
Gewicht z_i	2	3	2	3	1	2
Wert p_i	9	30	35	40	12	20

b) Beweise oder widerlege: In Algorithmus 1.15 kann die Schleife 2.1 auch in umgekehrter Reihenfolge durchlaufen werden, d.h. in der Reihenfolge $x = z_i - 1, x = z_i - 2, \dots, x = 0$.

(7+3 Punkte)

Aufgabe 2 (Longest Common Subsequence (LCS)): Betrachte das Problem Longest Common Subsequence aus der großen Übung.

Wende den in der großen Übung vorgestellten Algorithmus für LCS auf die Strings $BACCBABDDCB$ und $DBABCABDABA$ an, indem Du die folgende Tabelle ausfüllst. Der Eintrag in der i -ten Zeile und j -ten Spalte soll die Länge der längsten gemeinsamen Teilsequenz der Wörter beinhalten, die aus den ersten i bzw. ersten j Buchstaben bestehen.

$i \setminus j$	0	1	2	3	4	5	6	7	8	9	10	11	12
0													
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													

(6 Punkte)

Aufgabe 3 (LCS für drei Wörter): Entwerfe einen Algorithmus für LCS von drei Wörtern, welcher dynamische Programmierung verwendet. Beweise die Korrektheit Deines Algorithmus per vollständiger Induktion über die Längen der betrachteten Teilwörter der drei Eingabewörter.

(4 Punkte)