

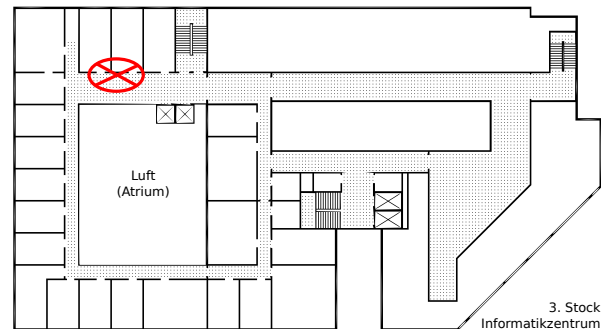
Prof. Dr. Sándor P. Fekete  
Dr. Christian Scheffer

## Algorithmen und Datenstrukturen II

### Übung 2 vom 04.05.2016

Abgabe der Lösungen bis zum Montag,  
den 23.05.2016 um 13:15 im Hausaufga-  
benrückgabeschrank.

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



**Aufgabe 1 (Subset Sum):** In dieser Aufgabe geht es um Algorithmus 1.11 (Dynamische Programmierung für Subset Sum) aus der Vorlesung.

a) Wir betrachten die folgenden Bestandteile von Algorithmus 1.11:

- (i) Die erste Zeile ( $\mathcal{S}(0,0) = 1$ ),
- (ii) die dritte Zeile ( $\mathcal{S}(x,0) = 0$ ),
- (iii) den IF...OR...ELSE-Block und
- (iv) die letzte Zeile (RETURN).

Wie oft werden i), ii), iii) und iv) jeweils ausgeführt?

b) Nimm an, dass die Kosten, die jeweils durch eine einmalige Ausführung von i), ii), iii) und iv) verursacht werden, durch die konstante Zahl  $c$  beschrieben sind. Nimm weiter an, dass die Ausführung aller restlichen Zeilen des Algorithmus keine Kosten verursachen. Welche Kosten verursacht der Algorithmus insgesamt? Beschreibe diesen Wert in Abhängigkeit von  $c$ ,  $n$  und  $Z$  und erkläre wie dieser zustande kommt.

c) Führe Algorithmus 1.11 bis  $x = 10$  für die folgende Eingabe aus:

$$\begin{aligned} \text{(a)} \quad Z_1 = 1, Z_2 = 2, Z_3 = 5, Z_4 = 3, Z_5 = 2, \\ Z_6 = 1, Z_7 = 2, Z_8 = 2, Z_9 = 4, Z_{10} = 5 \end{aligned}$$

Fülle hierzu folgende Tabelle aus (der Eintrag in der Zeile  $i$  und der Spalte  $x$  entspricht dem Wert  $\mathcal{S}(x, i)$ ):

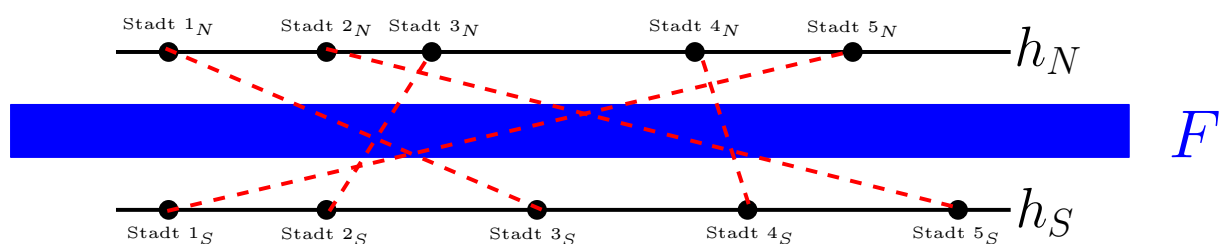
$i \backslash x$	0	1	2	3	4	5	6	7	8	9	10
0											
1											
2											
3											
4											
5											
6											
7											
8											
9											
10											

(2+1+5 Punkte)

**Aufgabe 2 (Brückenbau):** Betrachte einen Fluss  $F$ , der durch eine horizontale Gerade gegeben ist. Des Weiteren seien  $n$  Städte  $\{1_N, \dots, n_N\}$  gegeben, die auf einer Horizontalen  $h_N$  nördlich von  $F$  liegen. Analog seien  $n$  Städte  $\{1_S, \dots, n_S\}$  gegeben, die auf einer Horizontalen  $h_S$  südlich von  $F$  liegen, siehe zum Beispiel Abbildung 1. Jede Stadt im Norden hat genau eine Partnerstadt im Süden und anders herum. Eine Brücke zwischen einer Stadt im Norden und einer Stadt im Süden entspricht dem Segment zwischen beiden Städten.

Die Indizes zweier Partnerstädte müssen nicht übereinstimmen. Zum Beispiel kann  $2_N$  Partnerstadt von  $5_S$  sein.

- Entwirf einen Dynamic-Programming-Algorithmus, der die Kardinalität einer möglichst Großen Menge von kreuzungsfreien Brücken bestimmt, die jeweils zwei Partnerstädte miteinander verbinden.
- Wende Deinen Algorithmus auf die in Abbildung 1 dargestellte Städtesituation an, wobei folgende Partnerschaften gegeben seien:  $(1_N, 3_S)$ ,  $(2_N, 5_S)$ ,  $(3_N, 2_S)$ ,  $(4_N, 4_S)$  und  $(5_N, 1_S)$ .



**Abbildung 1:** Eine Städtekongfiguration. Die gestrichelten Linien deuten die Partnerschaften an.

(5+3 Punkte)

**Aufgabe 3 (Korrektheit von Algorithmen):** Betrachte Algorithmus 1. Was berechnet der Algorithmus? Beweise Deine Aussage mittels vollständiger Induktion über  $n$ . Gib

```
1: function ALGORITHMUS( $n$ )
2:   init  $DP[1..n]$ 
3:    $DP[0] = 1$ 

4:   for  $i = 1, \dots, n$  do
5:      $DP[i] \leftarrow 2 \cdot DP[i - 1]$ 
6:   end for

7:   return  $DP[n]$ 
8: end function
```

**Algorithmus 1:** Ein Algorithmus

alle Bestandteile einer vollständigen Induktion an. Gib explizit an, an welcher Stelle und wie Du die Induktionsvoraussetzung anwendest.

**(4 Punkte)**