technische universität dortmund

SoSe 2013

C. Sohler
A. Krivošija, H. Wagner

DAP2 – Heimübung 7

Ausgabedatum: 15.05.2013 — Abgabedatum: 24.05.2013

Abgabe:

Für Ihre Abgabe verwenden Sie bitte das beigefügte Blatt als Deckblatt. Fügen Sie, falls nötig, weitere Blätter hinzu. Schreiben Sie unbedingt immer Ihren vollständigen Namen, Ihre Matrikelnummer und Ihre Gruppennummer auf Ihre Abgaben!

Beachten Sie:

Beweise sind nur dort notwendig, wo explizit danach gefragt wird. Eine Begründung der Antwort wird allerdings *immer* verlangt.

Die Heimübungen dürfen in Gruppen von maximal drei Studierenden abgegeben werden. Die gemeinsame Bearbeitung in solchen Gruppen ist ausdrücklich erwünscht.

Aufgabe 7.1 (5 Punkte): Dynamische Programmierung

In der Vorlesung wurde der folgende Algorithmus `MaxDynamic` angegeben:

MaxDynamic(Array A):
$n \leftarrow Length(A)$
$M \leftarrow$ `new` $Array[Length(A)]$
$M[1] \leftarrow A[1]$
for $i = 2$ **to** $length(A)$ **do**
    $M[i] \leftarrow$ `max`$(M[i-1], A[i])$
return $M[n]$

Dieser Algorithmus ermittelt für ein Array A von reellen Zahlen das größte Element in diesem Array mittels dynamischer Programmierung.

Beweisen Sie die Korrektheit dieses Algorithmus.

Aufgabe 7.2 (5 Punkte): Dynamische Programmierung

Gegeben sei eine Kletterwand, die aus $n \times m$ Blöcken besteht. An jedem dieser Blöcke befindet sich eine Halterung, an der sich der Kletterer festhalten kann. Allerdings sind diese Halterungen zum Teil unsicher, so dass sie ein gewisses Gefährdungspotenzial aufweisen. Von einem Block ausgehend sind für den Kletterer nur die Blöcke erreichbar, die in der Zeile über seinem Block liegen und über die Ecke oder über eine Seite an seinen Block grenzen, d.h. die Blöcke, die links oben, oben oder rechts oben liegen. Der Kletterer ist bestrebt, eine Route vom unteren zum oberen Ende der Kletterwand zu finden, die das geringste Gefährdungspotenzial besitzt.

Diese Problemstellung werde über ein Array $C[1..n][1..m]$ von positiven ganzen Zahlen modelliert, wobei $C[i, j]$ das Gefährdungspotenzial der Halterung im Block (i, j) bezeichnet. Ein größerer Wert bedeutet dabei ein größeres Risiko. Im folgenden Beispiel wäre die rot gekennzeichnete Route die mit dem geringsten Gefährdungspotenzial.

8	4	1	6
1	2	2	5
2	7	4	2
1	2	5	4
6	2	4	1

1. Geben Sie zunächst einen rekursiven Algorithmus an, der das geringste Gefährdungspotenzial für den Kletterer berechnet.
2. Geben Sie dann (in Pseudocode) einen auf dynamischer Programmierung beruhenden Algorithmus an, der das geringste Gefährdungspotenzial bestimmt.
3. Führen Sie eine exakte Laufzeitanalyse für Ihren Algorithmus durch.
4. Beweisen Sie die Korrektheit Ihres Algorithmus.

Aufgabe 7.3 (5 Punkte): Dynamische Programmierung

Gegeben sei ein Array $A[1..n]$ von ganzen Zahlen. Unter allen Summen $\sum_{k=i}^{j} A[k]$ von Teilarrays $A[i..j]$ mit $1 \leq i \leq j \leq n$ wollen wir diejenige bestimmen, die minimal ist. Z.B. besitzt in dem Array $[3, 1, -4, -2, 1, -3, 4, -2, 5]$ das Teilarray $[-4, -2, 1, -3]$ die minimale Summe -8.

1. Geben Sie einen rekursiven Algorithmus an, der bei Eingabe eines Arrays $A[1..n]$ die minimale Summe berechnet.
2. Geben Sie dann (in Pseudocode) einen auf dynamischer Programmierung beruhenden Algorithmus an, der in Zeit $O(n)$ diese minimale Summe bestimmt.
3. Beweisen Sie die Korrektheit Ihres Algorithmus.
4. (**Bonusaufgabe (4 Punkte)**) Geben Sie einen Teile-und-Herrsche Algorithmus für die Berechnung der minimalen Summe an.