

DAP2 – Heimübung 8

Ausgabedatum: 22.05.2013 — Abgabedatum: 31.05.2013

Abgabe:

Für Ihre Abgabe verwenden Sie bitte das beigegefügte 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 8.1 (5 Punkte): Dynamische Programmierung

Gegeben sei ein Array $A[1..n]$ von ganzen Zahlen. Ein strikt unzusammenhängendes Teilarray von $A[1..n]$ ist ein Array $[A[i_1], A[i_2], \dots, A[i_k]]$ mit der Eigenschaft $1 \leq i_1 < i_1 + 1 < i_2 < i_2 + 1 < \dots < i_{k-1} + 1 < i_k \leq n$, das bedeutet, dass keine zwei in A direkt benachbarte Elemente in B zusammen auftreten können.

Unter allen strikt unzusammenhängenden Teilarrays B von $A[1..n]$ wollen wir eines bestimmen, für das die Summe seiner Elemente maximal ist.

Z.B. hat in dem Array $[3, 6, 4, -2, 1, -3, 4, 8, 2]$ das strikt unzusammenhängende Teilarray $[3, 4, 1, 8]$ die maximale Summe 16.

1. Geben Sie einen rekursiven Algorithmus an, der bei Eingabe eines Arrays $A[1..n]$ ein strikt unzusammenhängendes Teilarray B von A berechnet, für das die Summe seiner Elemente maximal ist.
2. Geben Sie dann (in Pseudocode) einen auf dynamischer Programmierung beruhenden Algorithmus an, der dieses Array B bestimmt.
3. Beweisen Sie die Korrektheit Ihres Algorithmus.
4. Geben Sie eine möglichst kleine obere Schranke für die Worst-Case-Laufzeit Ihres Algorithmus in O -Notation an und begründen Sie diese.

Aufgabe 8.2 (5 Punkte): Dynamische Programmierung

Ein Dieb ist in eine Bank eingebrochen. Die Bank ist ein sehr hohes Gebäude mit 150 oberirdischen Etagen. Momentan befindet sich der Dieb im Erdgeschoss, das als Geschoss 0 der Bank bezeichnet wird. In der Kasse des i -ten Geschosses, $1 \leq i \leq n$, ist ein Betrag von A_i Euro vorhanden, was dem Dieb bekannt ist.

Zur gleichen Zeit entzündet sich ein Brand, wodurch sich alle Kassen öffnen. Es ist Nacht, daher befinden sich keine Beschäftigte in dem Gebäude. Die Brandursache befindet sich in der k -ten Etage und der Brand breitet sich mit einer Geschwindigkeit von 1 Etage pro Minute in alle Richtungen aus. Der Dieb muss sich natürlich vor dem Feuer hüten und darf daher zu keinem Zeitpunkt auf einer Etage sein, auf der das Feuer schon ausgebrochen ist.

In dem Gebäude gibt es einen Fahrstuhl, der 10 Etagen pro Minute fährt, aber auf einer Etage, auf der das Feuer bereits ausgebrochen ist, stoppt. Am Anfang befindet sich der Fahrstuhl im Erdgeschoss. Um das Geld in einer Etage zu finden und es zum Fahrstuhl mitzubringen, benötigt der Dieb 1.5 Minuten.

Wir suchen einen Algorithmus, der für eine gegebene Folge $A[1..150]$ von Beträgen und der Nummer k der Etage, auf der das Feuer ausgebrochen ist, den größten Betrag $B(k)$, den der Dieb als Beute erzielen kann, berechnet.

1. Geben Sie zunächst einen rekursiven Algorithmus an, der die größte Beute $B(k)$, die für den Dieb möglich ist, berechnet.
2. Geben Sie dann (in Pseudocode) einen auf dynamischer Programmierung beruhenden Algorithmus an, der den in Teilaufgabe 1. berechneten Beutebetrag bestimmt.
3. Beweisen Sie die Korrektheit Ihres Algorithmus.

Hinweis: Berechnen Sie den Wert der Beute als Funktion vom Zeitpunkt t und vom Geschoss i , in dem sich der Dieb befindet.

DAP2 – Heimübung 8

| Name: | Vorname | Matr.Nr.: | Gr.Nr.: |
|-------|---------|-----------|---------|
| | | | |
| | | | |
| | | | |
