

DAP2 – Heimübung 8

Ausgabedatum: 2.6.17 — Abgabedatum: Fr. 9.6.17 (Mo. 12.6. für Gruppen 27–32) 12 Uhr

Schreiben Sie unbedingt immer Ihren **vollständigen Namen**, **Ihre Matrikelnummer** und **Ihre Gruppennummer** auf Ihre Abgaben!

Aufgabe 8.1 (5 Punkte): (Dynamische Programmierung)

In einer Uni im Ausland gelten folgende Regeln für das Programmierpraktikum: es gibt n Übungsblätter, und i -tes Übungsblatt ist $A[i]$ Punkte wert, $1 \leq i \leq n$. Wenn man aber die Punkte im Blatt i gekriegt hat, kann man weder im Blatt $i - 1$ noch im Blatt $i + 1$ Punkte kriegen. Alle Werte $A[i]$ sind positive ganze Zahlen.

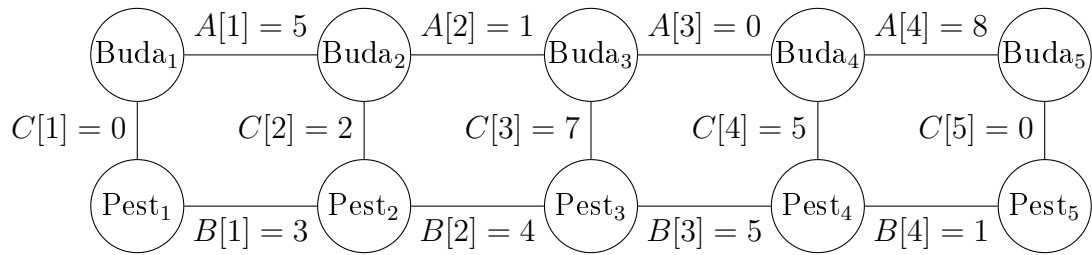
Der Student Bob möchte wissen, wie viele Punkte kann er unten diesen Regeln sammeln. Z.B. wenn die Punktwerte aller Übungsblätter im Array $A = [3, 6, 5, 1, 1, 8, 4, 2]$ gegeben sind, ist es maximal möglich 18 Punkte zu sammeln, wenn man die Übungsblätter mit $[3, 5, 8, 2]$ Punkte bearbeiten würde.

- Geben Sie eine Rekursionsgleichung für die maximale Anzahl der erreichbaren Punkte $B[i]$, wenn man nur die Übungsblätter $A[1..i]$ betrachtet.
- Geben Sie dann (in Pseudocode) einen auf dynamischer Programmierung beruhenden Algorithmus an, der die maximal erreichbare Punktesumme des Arrays $A[1..n]$ bestimmt und zurückgibt.
- Analysieren Sie die Laufzeit Ihres Algorithmus.
- Beweisen Sie die Korrektheit Ihres Algorithmus.

Aufgabe 8.2 (5 Punkte): (Dynamische Programmierung)

In Budapest gibt es n Brücken am Donau. Alice möchte die Strecke zwischen Brücken 1 und n am Ufer entlang spazieren, wobei sie zwischen zwei nacheinanderliegenden Brücken entweder am rechten Ufer (in Buda) oder am linken Ufer (in Pest) spazieren kann. Während sie zwischen Brücken i und $i + 1$ spaziert, $1 \leq i < n$, kriegt sie Brezeln, und zwar $A[i]$ Brezeln in Buda, und $B[i]$ Brezeln in Pest. Wenn sie Donau über i -te Brücke überqueren möchte, muss sie $C[i]$ Brezeln abgeben, wobei sie nie weniger als 0 Brezeln haben kann.

Alice beginnt mit 0 Brezeln in Buda bei der Brücke 1. Alle Werte $A[i]$ und $B[i]$ ($1 \leq i \leq n - 1$), sowie $C[i]$ ($1 \leq i \leq n$) sind nicht negativ, und es gilt $C[1] = C[n] = 0$. Alice darf nicht in Gegenrichtung laufen (z.B. von Brücke $i + 1$ nach Brücke i) und will die Anzahl der gesammelten Brezeln bei der n -ten Brücke in Buda maximieren.



In diesem Beispiel wäre es optimal, bei den Brücken 2 und 4 das Ufer zu wechseln. In diesem Fall kommt man mit $15 (= 5 - 2 + 4 + 5 - 5 + 8)$ Brezeln am Ziel in Buda an.

- a) Geben Sie die Rekursionsgleichungen für die maximalen Anzahlen $p(i)$ und $q(i)$ von Brezeln an, die Alice bei Erreichen der i -ten Brücke in Buda bzw. Pest sammeln kann.
- b) Geben Sie einen Algorithmus in Pseudocode an, der auf dem Prinzip der dynamischen Programmierung beruht und die maximale Anzahl gesammelter Brezeln beim Erreichen der Brücke n in Buda zurückgibt.
- c) Analysieren Sie die Laufzeit Ihres Algorithmus.
- d) Beweisen Sie die Korrektheit Ihrer Rekursionsgleichungen aus Teilaufgabe a) mittels Induktion.