

## DAP2 – Heimübung 8

Ausgabedatum: 22.05.2018 — Abgabedatum: Mo. 04.06.2018 bis 12 Uhr

### **Aufgabe 8.1 (5 Punkte):** (Dynamische Programmierung: Geschickter Wahlkampf)

In  $k$  Tagen findet eine Wahl statt. Bob kann an Tag  $i$  mit Wahlwerbung  $a_i$  neue Wähler gewinnen. Allerdings ist das Werben von Wählern so anstrengend, dass er spätestens nach je zwei Tagen einen Tag Pause braucht. Sein Ziel ist es, so viele neue Wähler wie möglich zu gewinnen.

Wir betrachten ein Array  $A$  der Länge  $k$ , in dem die Werte  $A[i] = a_i > 0$ ,  $1 \leq i \leq k$ , gespeichert sind.

Beispielsweise sei  $A = [2, 3, 4, 5, 6]$ , dann kann Bob am 1., 2., 4. und 5. Tag die maximale Anzahl, nämlich 16 neue Wähler, erreichen.

- Es bezeichne  $D(i, j)$ ,  $1 \leq i \leq k$ ,  $j \in \{0, 1\}$ , die maximale Anzahl an neuen Wählern, die Bob an den ersten  $i$  Tagen erreichen kann, wenn er entweder Tag  $i$  hinzunimmt ( $j = 1$ ) oder an Tag  $i$  eine Pause macht ( $j = 0$ ). Geben Sie eine rekursive Form für  $D(i, j)$  an.
- Geben Sie einen Algorithmus in Pseudocode an, der auf dem Prinzip der dynamischen Programmierung beruht und die maximale Anzahl an neuen Wählern zurückgibt, die Bob gewinnen kann.
- Analysieren Sie die Laufzeit Ihres Algorithmus.
- Beweisen Sie die Korrektheit Ihrer rekursiven Form aus Teilaufgabe a) mittels Induktion.

### **Aufgabe 8.2 (5 Punkte):** (Dynamische Programmierung: Sparen mit Bonuspunkten)

Alice möchte gern das neue Pfannenset kaufen, das es im Supermarkt für 50 Bonuspunkte günstig zu erwerben gibt. Leider hat sie aktuell keine Bonuspunkte. Im Supermarkt gibt es  $n$  Produkte zu kaufen. Produkt  $i$  bringt  $b_i$  Bonuspunkte und kostet  $p_i$  Euro. Zusätzlich gibt es eine Beschränkung, dass alle gekauften Produkte unterschiedlich sein müssen. Alice möchte jetzt herausfinden, wie teuer es für sie mindestens wird, 50 Bonuspunkte zu sammeln.

Entwerfen Sie einen Algorithmus, der die Kosten einer günstigsten Auswahl von Produkten ermittelt, sodass Alice mindestens 50 Bonuspunkte erhält.

- Wir bezeichnen mit  $A(i, j)$  die minimalen Kosten, die Alice bezahlen muss, um mit einer Auswahl aus den ersten  $i$  Produkten mindestens  $j$  Bonuspunkte zu erhalten. Geben Sie eine rekursive Form für  $A(i, j)$  und alle Werte  $i = 0, \dots, n$  und  $j = 0, \dots, 50$  an. Denken Sie insbesondere an die Basisfälle.

- b) Geben Sie einen Algorithmus in Pseudocode an, der das Problem basierend auf der rekursiven Form mit dynamischer Programmierung löst.
- c) Analysieren Sie die Laufzeit Ihres Algorithmus.
- d) Beweisen Sie die Korrektheit Ihrer rekursiven Form aus Teilaufgabe a), zeigen Sie also, dass  $A(i, j)$  nach Ihrer Formel wirklich den optimalen Wert für das dort beschriebene Teilproblem enthält.