Prof. Dr. Christian Sohler
Nicole Funk, Dr. Matthijs Ebbens, Sebastian Zaun

Sommersemester 2023

6. Übungsblatt

zur Vorlesung

Grundzüge der Informatik I

Abgabe über Ilias bis zum 17.4. 14:00 Uhr.
Besprechung in Kalenderwoche 21.

Aufgabe 1 *(4 Punkte)*

Wir betrachten den Algorithmus $SubsetSum(A, U, n)$ der Vorlesung. Gegeben sei die folgende Menge

$$A = \{19, 5, 7, 4, 12, 9, 2, 6, 4\} \tag{1}$$

und der Wert

$$u = 15 \tag{2}$$

Wenden Sie den Algorithmus der Vorlesung an, um zu entscheiden, ob es eine Teilmenge $L \subseteq A$ gibt, sodass $\sum_{x \in L} x = u$ gilt. Geben Sie dabei das vollständige Array Ind, sowie das Ergebnis des Algorithmus an.

Aufgabe 2 *(4 Punkte)*

Gegeben sei ein Array $A[1..n]$ mit $1 \leq A[i] \leq 3$ für alle $1 \leq i \leq n$. Eine Spielfigur startet auf der ersten Stelle des Arrays und muss die n-te Stelle erreichen. Befindet sich die Figur auf der Stelle i für $1 \leq i \leq n$, so darf sie mit einem Sprung bis zu $A[i]$ Stellen nach vorne ziehen. Im unten gezeigten Beispiel darf die Figur also von der zweiten Stelle aus bis zu $A[2] = 3$ Stellen weiterspringen, also jede der Stellen $3, 4$ und 5 mit einem Sprung erreichen. Gesucht ist die minimale Anzahl von Sprüngen, um beginnend auf der ersten Stelle des Arrays die n-te Stelle zu erreichen.

Für das folgende Beispiel mit $n = 8$ beträgt die minimale Anzahl an Sprüngen 3 und ergibt sich durch die Sprungfolge $1 \curvearrowright 2 \curvearrowright 5 \curvearrowright 8$.

$$A = \begin{pmatrix} \mathbf{2} & \mathbf{3} & 1 & 2 & \mathbf{3} & 1 & 2 & \mathbf{1} \end{pmatrix}$$

Sei $M[i]$ die minimal benötigte Anzahl von Sprüngen, um ausgehend von der i-ten Stelle die n-te Stelle zu erreichen. Geben Sie eine rekursive Formulierung für $M[i]$ an. Erklären Sie die Funktionsweise dieser. Gehen Sie dabei auf jede Fallunterschiedung ein.

Aufgabe 3 *(4 + 2 + 4 + 2 Punkte)*

Gegeben sei eine Menge A mit n Zahlen. Die Anzahl der Partitionen von A kann mit der sogenannten Stirling Zahl zweiter Art berechnet werden.

Diese Zahl $S(n, k)$ ist mit $k, n \in \mathbb{N}_0$ und $n \geq k$ rekursiv definiert als:

$$S(n,k) = \begin{cases} 0 & \text{falls } k = 0, n > 0 \\ 1 & \text{falls } k = n \\ S_{n-1,k-1} + k \cdot S_{n-1,k} & \text{sonst.} \end{cases} \tag{3}$$

und beschreibt wie viele Partitionen einer n elementigen Menge in k disjunkte Teilmengen es gibt.

a) Geben Sie einen rekursiven Algorithmus in Pseudocode an, welcher bei Eingabe einer Zahl n unter Verwendung von $S(n, k)$ die Anzahl aller Partitionen einer Menge mit n Elementen berechnet.

b) Analysieren Sie die asymptotische Worst-Case-Laufzeit Ihres Algorithmus aus Teilaufgabe a).

c) Geben Sie einen Algorithmus in Pseudocode an, der auf dem Prinzip der dynamischen Programmierung beruht, und bei Eingabe einer Zahl n unter Verwendung von $S(n, k)$ die Anzahl aller Partitionen einer Menge mit n Elementen berechnet.

d) Analysieren Sie die asymptotische Worst-Case-Laufzeit Ihres Algorithmus aus Teilaufgabe c).