Aufgabe 1: LaTeX-Dokument

Teilnahme-ID: ?????

Bearbeiter/-in dieser Aufgabe:
Vor- und Nachname

2. Februar 2023

Inhaltsverzeichnis

Anleitung: Trage oben in den Zeilen 8 bis 10 die Aufgabennummer, die Teilnahme-ID und die/den Bearbeiterin/Bearbeiter dieser Aufgabe mit Vor- und Nachnamen ein. Vergiss nicht, auch den Aufgabennamen anzupassen (statt „LaTeX-Dokument“)!

Dann kannst du dieses Dokument mit deiner LaTeX-Umgebung übersetzen.

Die Texte, die hier bereits stehen, geben ein paar Hinweise zur Einsendung. Du solltest sie aber in deiner Einsendung wieder entfernen!

1 Lösungsidee

1.1 Notation

Die Menge der möglichen Stapel der Höhe n wird mit $\mathcal{P}_n$ bezeichnet. Die Möglichen Pfannkuchen-Wende-und-Ess-Operationen für einem Stapel mit n Pfannkuchen wird mit $\mathcal{W}_n$ bezeichnet. Die Menge der möglichen Umkehroperationen für solch einen Stapel wird mit $\mathcal{W}_n^{-1}$ bezeichnet. Wenn der Stapel $S \in \mathcal{P}_n$ durch die Operation $w \in \mathcal{W}_n$ verändert wird, so wird der neue Stapel $S' = wS$ bezeichnet. Operationen assoziieren nach rechts, d.h. $w_1 w_2 S = w_1(w_2 S)$. Die Funktionen A und P werden aus der Aufgabenstellung übernommen.

1.2 PWUE-Zahl

Die PWUE-Zahl kann rekursiv mit Hilfe der dynamischen Programmierung berechnet werden. Dafür definieren wir die Funktion $K(n,a) = \{s \in \mathcal{P}_n \mid A(s) = a\}$, die die Menge aller Stapel der Höhe n enthält, die in mindestens a Schritten sortiert werden können. Die Funktion lässt sich rekursiv berechnen:

$$\begin{aligned}
K(n,a) &= \{ws, w \in \mathcal{W}_{n-1}^{-1}, s \in K(n-1, a-1)\} | \forall v \in \mathcal{W}_n : A(vws) \geq a-1\} \\
&= \{ws, w \in \mathcal{W}_{n-1}^{-1}, s \in K(n-1, a-1)\} | \forall v \in \mathcal{W}_n : \exists b \geq a-1 : A(vws) = b\} \\
&= \{ws, w \in \mathcal{W}_{n-1}^{-1}, s \in K(n-1, a-1)\} | \forall v \in \mathcal{W}_n : \exists b \geq a-1 : vws \in K(n-1, b)\}
\end{aligned}$$

$K(n, a)$ enthält also alle Stapel, die durch eine Umkehroperation aus Stapeln der Höhe $n-1$ mit mindestens $a-1$ Sortieroperationen entstehen können und für die keine andere Sortieroperation eine Stapel bildet, der in weniger als $a-1$ Schritten sortiert werden kann. Nach dieser Definition würde $K(n, 1)$ allerdings auch die komplett sortierten Stapel enthalten, weshalb noch die Bedingung $(a > 1) \vee (s \notin K(n, 0))$ ergänzt werden muss. Die Funktion $K(n, a)$ ist also definiert als

$$K(n,a) = \{ws, w \in \mathcal{W}_{n-1}^{-1}, s \in K(n-1, a-1)\} | \forall v \in \mathcal{W}_n : \exists b \geq a-1 : vws \in K(n-1, b) \wedge ((a > 1) \vee (s \notin K(n, 0)))\}$$

Dass diese Definition richtig ist, lässt sich überprüfen durch die Substitution $A(S) = k, S \in \mathcal{P}_n \iff (\forall w \in \mathcal{W}_n : A(ws) \geq k-1) \wedge (\exists w \in \mathcal{W}_n : A(ws) = k-1)$. Um jetzt die PWUE-Zahl zu berechnen, muss nur noch die Funktion $K(n, a)$ für alle a berechnet werden und überprüft werden, ob sie Elemente enthält. Da jeder Stapel $S \in \mathcal{P}_n$ in $\lceil \frac{n}{1.5} \rceil$ Schritten sortiert werden kann, reicht es aus, die Funktion $K(n, a)$ für alle $\lceil \frac{n}{1.5} \rceil$ zu berechnen. Damit lässt sich auch $\exists b \geq a-1 : vws \in K(n-1, b)$ durch $\exists \lceil \frac{n}{1.5} \rceil \geq b \geq a-1 : vws \in K(n-1, b)$ ersetzen wodurch nicht unendlich viele Werte für b ausprobiert werden müssen. Zuletzt muss noch ein Ende der Rekursion eingeführt werden, wir setzen $K(n, 0) = \{(1, \ldots, n)\}$.

2 Umsetzung

Hier wird kurz erläutert, wie die Lösungsidee im Programm tatsächlich umgesetzt wurde. Hier können auch Implementierungsdetails erwähnt werden.

3 Beispiele

Genügend Beispiele einbinden! Die Beispiele von der BwInf-Webseite sollten hier diskutiert werden, aber auch eigene Beispiele sind sehr gut – besonders wenn sie Spezialfälle abdecken. Aber bitte nicht 30 Seiten Programmausgabe hier einfügen!

4 Quellcode

Unwichtige Teile des Programms sollen hier nicht abgedruckt werden. Dieser Teil sollte nicht mehr als 2–3 Seiten umfassen, maximal 10.