

J1 Bücherregal

J1.1 Lösungsidee

Bei dieser Aufgabe soll eine Menge von Büchern in Abschnitte aufgeteilt werden, in denen die Bücher etwa gleich groß sind. „Etwa gleich groß“ bedeutet im Sinne der Aufgabe, dass keine zwei Bücher in einem Abschnitt einen größeren Höhenunterschied haben dürfen als 3 Zentimeter.

Da nur eine bestimmte Anzahl an Deko-Figuren zur Verfügung steht, um die Abschnitte voneinander abzutrennen, ist die Anzahl der möglichen Abschnitte begrenzt: Es darf höchstens einen Abschnitt mehr geben als Deko-Figuren vorhanden sind; wenn D die Zahl der Deko-Figuren ist, darf es also höchstens $D + 1$ Abschnitte geben. Die Frage ist nun, ob eine Einteilung in Abschnitte unter diesen Bedingungen möglich ist und, falls ja, wie. Dabei wollen wir erlauben, dass Abschnitte leer bleiben. In der Aufgabenstellung heißt es nämlich: „Wie breit die Abschnitte sind, ist ihr egal.“ Also sollte auch eine Breite von 0 akzeptabel sein.

Diese Annahme erlaubt eine besonders einfache Lösungsstrategie: Man wählt zunächst das kleinste Buch aus, das noch nicht im Regal steht, und fügt es mit allen Büchern in einem Abschnitt zusammen, die nicht mehr als 3 Zentimeter größer sind als dieses Buch. Eine derartige Strategie heißt in der Informatik auch „greedy“ (engl. für: gierig), weil sie sich gierig alle Bücher nimmt, die in den Abschnitt passen, ohne zu beachten, ob sie auch in einen anderen Abschnitt passen würden. Dennoch entsteht auf diese Weise eine Einteilung in möglichst wenige Abschnitte; die Strategie ist also optimal in Bezug auf die Anzahl der verwendeten Abschnitte. Das ist aber hilfreich: Nur wenn die Lösung verlässlich die kleinste Zahl von Abschnitten finden kann, kann sie sicher feststellen, ob eine passende Einteilung der Buchmenge evtl. nicht möglich ist.

Die Strategie nimmt allerdings in Kauf, dass weniger als $D + 1$ Abschnitte entstehen und einige Deko-Figuren nicht als Trenner verwendet werden. Wer das so nicht haben möchte, kann diese Situation „reparieren“ und übrige Deko-Figuren einfach irgendwo mitten in je einen gefundenen Abschnitt platzieren. Da jeder gefundene Abschnitt die Höhenbedingung erfüllt, wird die Bedingung auch von jedem seiner Teile erfüllt.

J1.2 Umsetzung

Die oben beschriebene Strategie lässt sich auf verschiedene Weise in einem Programm umsetzen. Werden die Bücherhöhen in eine Liste geschrieben, kann man die Liste sortieren (in vielen Programmiersprachen gibt es eine Sortierfunktion) und hat das gesuchte kleinste Buch vorne in der Liste stehen. Die sortierte Liste kann dann einfach der Reihe nach durchgegangen werden, bis ein Buch mehr als 3cm höher ist als das erste. Dieses Buch ist dann das erste Buch eines neuen Abschnitts, und alle Bücher bis dahin gehören noch zum vorigen Abschnitt. Insgesamt muss man die sortierte Liste einmal durchlaufen.

Sortieren lässt sich eine Menge aber auch, indem man ihre Elemente nach und nach in eine Datenstruktur einfügt, die am Ende ihre Elemente sortiert ausgeben kann. Auch solche Datenstrukturen gibt es in vielen Programmiersprachen; in Java z. B. `SortedSet` bzw. `TreeSet`.

Beispiel	# 1	# 2	# 3	# 4	# 5	# 6
Gegeben	4 Figuren	2 Figuren	2 Figuren	4 Figuren	3 Figuren	4 Figuren
Benötigt	4 Figuren	2 Figuren	3 Figuren	4 Figuren	3 Figuren	5 Figuren
Ergebnis	Möglich	Möglich	Unmöglich	Möglich	Möglich	Unmöglich
Abschnitte	168	169	170	160	160	125
	170	175	174	:	:	Figur
	Figur	Figur	Figur	190	180	160
	202	203	202	Figur	Figur	:
	211	209	229	196	201	180
	229	210	Figur	:	:	Figur
	Figur	229	235	225	231	200
	233	Figur	Figur	Figur	Figur	:
	254	235	279	233	232	228
	260		305	:	:	Figur
	Figur			261	261	247
	272			Figur	Figur	:
	Figur			264	263	264
	306			:	:	Figur
	307			293	270	281
				Figur		:
				295		310
				:		Figur
				304		340
						341
						342

Tabelle 1: Ergebnisse zu den vorgegebenen Beispielen

J1.3 Alternative Lösungsideen

Man kommt zur Not auch ohne Sortieren zurecht. Dazu muss man allerdings für jeden Abschnitt zweimal die gesamte Buchmenge durchsuchen: einmal, um das kleinste Buch zu finden und noch einmal, um alle Bücher herauszusuchen, die nicht mehr als 3cm höher sind. Ist N die Anzahl der Bücher und (wie schon gesagt) D die Anzahl der Deko-Figuren, muss man also $2(D+1)$ mal maximal N Bücher anschauen. Allerdings braucht bei ersten Strategie das Sortieren auch seine Zeit.

J1.4 Beispiele

Die Ergebnisse, die unsere „gierige“ Strategie für die vorgegebenen Beispiele liefert, sind in der Table 1 zusammengefasst. Anders als in der Aufgabenstellung gesagt wurden die Buchgrößen in den Beispieldateien in Millimetern angegeben, nicht in Zentimetern.

Diese Ergebnisse müssen nicht exakt so reproduziert werden. Z. B. ist auch ein Sortieren in absteigender Reihenfolge möglich und ergibt dann z. T. andere Verteilungen. Auch die Reihenfolge der Abschnitte oder die Reihenfolge der Bücher innerhalb der Abschnitte spielen keine Rolle.

J1.5 Bewertungskriterien

- Der Lösungsansatz (und das Programm) erfüllen die genannten Bedingungen:
 - Der Höhenunterschied beträgt pro Abschnitt höchstens 3 cm, und
 - es werden nicht mehr Abschnitte erzeugt als durch Figuren getrennt werden können.
- Die Lösung muss die minimale Anzahl an Abschnitten finden können, um auch unmögliche Fälle sicher zu bestimmen.
- Das Programm gibt die Bücherhöhen der Abschnitte aus. Es genügt auch, für jeden Abschnitt die Höhen des kleinsten und des größten Buchs eines Abschnitts anzugeben.
- Die Lösung sollte mit einer Laufzeit von höchstens $\mathcal{O}(n^2)$ auskommen.
- Die vorgegebenen Beispiele sollten bearbeitet und die zugehörigen Ergebnisse dokumentiert sein.