Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Thomas Seidl
Janina Sontheim, Florian Richter

München, 25.07.2019

Algorithmen und Datenstrukturen
SS 2019

Übungsblatt Global 7: Paradigmen

Aufgabe Global 7-1 *Knobelei: Piratenschatz aufteilen*

Ein Piratenkapitän hat 100 Münzen, die er auf sich und seine 4 Personen starke Mannschaft aufteilen muss. Da Piraten demokratisch organisiert sind, darf jeder bei der Verteilung mitbestimmen. Der Kapitän schlägt eine Aufteilung vor. Ist die Mehrheit dafür (bei Unentschieden wiegt die Stimme des Kapitäns mehr), dann werden die Münzen verteilt. Ansonsten geht der Kapitän über die Planke und der Nächste in der Rangfolge wird neuer Kapitän. Die Prozedur wird wiederholt. Jeder Pirat handelt nach folgenden Motiven:

1. Jeder Pirat hat als oberstes Ziel, am Leben zu bleiben.
2. Zweitens möchte jeder seinen Gewinn maximieren.
3. Jeder Pirat lässt seinen Kapitän über die Planke gehen, falls alles andere keinen Unterschied macht.

Jeder Pirat hat die gleiche Zielsetzung und weiß auch, dass dies für die anderen gilt. Piraten können in keiner Weise Absprachen treffen oder sonst zusammen arbeiten. Keine Tricks, Täuschungen, Bestechungen oder sonstige Kommunikation außerhalb der Abstimmungen. Jeder Pirat ist ein ausgezeichneter Logiker. Wie verteilt der Kapitän die Münzen?

Lösungsvorschlag:

Wir erinnern uns an das Divide-and-Conquer-Prinzip und betrachten erst mal einen kleinen Fall: 1 Pirat E hat nichts zu befürchten. Er behält also 100 Münzen und ist glücklich. Die Strategie ist (100).
Für zwei Piraten D,E muss D eine Teilung vorschlagen. Da D den untergebenen E überstimmen kann, behält D also 100 Münzen und E kann nichts dagegen tun. Strategie: $(100, 0)$
Bei drei Piraten C,D,E braucht C unbedingt die Unterstützung eines weiteren Piraten. Er weiß: D wird ihn definitiv über die Planke laufen lassen, da D als Kapitän alle Münzen selbst behalten darf. E allerdings würde dann nichts bekommen. Also muss er E 1 Münze geben, was mehr ist als E erhalten würde, wenn D Kapitän wird, und behält selbst 99 Münzen. $(99, 0, 1)$
Nun gibt es vier Piraten B,C,D,E. B benötigt auch die Unterstützung eines anderen Piraten. Auf C kann er kaum hoffen, denn ihm müsste er mehr als 99 Münzen geben. E ist auch ein schlechter Verbündeter, denn der bekommt eine Münze, falls C Kapitän werden würde. D allerdings geht leer aus, falls C Kapitän wird. Daher reicht D eine Münze, um seine Gunst zu gewinnen. $(99, 0, 1, 0)$
Betrachten wir alle 5 Piraten A,B,C,D,E, dann benötigt A die Unterstützung von 2 anderen Piraten. Falls B Kapitän wird, dann gehen C und E leer aus. Beiden jeweils eine Münze zu geben, ist daher die optimale Strategie, um den eigenen Gewinn zu maximieren. $(98, 0, 1, 0, 1)$

Aufgabe Global 7-2 *Münzen und dynamische Programmierung*

Gegeben seien n verschiedene, positive Münzwerte $a_1, ..., a_n$ und ein Betrag m. Gesucht ist die Zahl der möglichen n-Tupel $(k_1, ..., k_n)$ mit $\sum_{l=1}^{n} k_l a_l = m$. Dies entspricht der Zahl der Möglichkeiten, m durch die gegebenen Münzwerte darzustellen. Nehmen Sie an, dass beliebig viele Münzen von jeder Sorte zur Verfügung stehen und sich der Betrag $m = 0$ durch genau einmal 0 Münzen darstellen lässt. Dieses Problem lässt sich in tabellarischer Form mittels dynamischer Programmierung darstellen. Dazu definieren wir Teilprobleme $t_{i,j}$, wobei i die Anzahl der Münzwerte $(0 <= i <= n)$ sind und j der Betrag $(0 <= j <= m)$ ist. Die Lösung des Gesamtproblems ist dann $t_{n,m}$. Die folgende Rekursionsgleichung gilt für $i > 0$:

$$t_{i,j} = t_{i-1,j} + \begin{cases} t_{i,j-a}, & j - a_i >= 0 \\ 0, & \text{sonst} \end{cases}$$

Geben Sie für $n = 3, m = 10, a_1 = 2, a_2 = 3, a_3 = 5$ die Matrix $[t_{i,j}]$ an. Geben Sie zu dem Ergebnis in $t_{n,m}$ die verschiedenen Möglichkeiten, 10 darzustellen, an. (Achtung: Was ist der Basisfall, von dem der Aufbau der Matrix startet?)

Lösungsvorschlag:

	0	1	2	3	4	5	6	7	8	9	10
0	1	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	1	0	1	0	1	0	1
2	1	0	1	1	1	1	2	1	2	2	2
3	1	0	1	1	1	2	2	2	3	3	4

Aufgabe Global 7-3 *Drei Steine in der Ecke*

Gegeben sei ein tabellenartiges Spielfeld, dass sich unendlich weit nach oben und rechts erstreckt. In der Ecke und rechts daneben sowie darüber befinden sich ingesamt drei Steine. Man kann einen Spielstein entfernen und dafür einen Spielstein darüber und einen weiteren auf das Feld rechts legen, falls beide Felder frei sind. Geben Sie einen Algorithmus an, der alle drei anfangs belegten Felder leert.

Lösungsvorschlag:

Nach einigen Versuchen stellen wir fest, dass es schwieriger zu lösen ist, als intuitiv ersichtlich ist. In solch einem Fall ist es oft sinnvoll, Invarianten zu identifizieren und so das Problem und die möglichen Operationen besser zu verstehen. Dazu nummerieren wir alle Felder, beginnend mit dem Feld links unten. Das erste Feld wird mit 1 beschriftet. Die angrenzenden Felder mit 1/2, die daran angrenzenden Felder mit 1/4 und so weiter.

1/32	1/64	1/128	1/256	1/512	1/1024
1/16	1/32	1/64	1/128	1/256	1/512
1/8	1/16	1/32	1/64	1/128	1/256
1/4	1/8	1/16	1/32	1/64	1/128
1/2	1/4	1/8	1/16	1/32	1/64
1	1/2	1/4	1/8	1/16	1/32

Zu Beginn liegen alle Steine auf den drei Feldern unten links. Die Summe der abgedeckten Feldwerte beträgt $1+1/2+1/2=2$. Bewegt man einen Stein, so wird der Wert x auf diesem Feld frei. Dafür legen wir Steine auf Felder, die nach der Beschriftungsvorschrift in der Summe wieder den Wert x haben. Dies ist unsere Invariante, denn egal welche Operationen wir in welcher Reihenfolge ausführen, die abgedeckte Summe beträgt stets 2.

Nun summieren wir alle Feldsummen des gesamten unendlichen Spielfelds auf. Klingt erst schwierig, aber wir sehen bald, dass die Beschriftung dies ohne Probleme ermöglicht. Wir summieren erst die unterste Reihe auf: $1+1/2+1/4+1/8+\ldots=\sum_{i=1}^{\infty}(1/2)^i=2$. Die Reihe darüber beginnt ein Summenglied später, daher hat diese Reihe den Wert 1. In der nächsten Reihe beträgt die Summe 1/2 usw. Alle Reihen können wir damit auch aufsummieren: $2+1+1/2+1/4+1/8+\ldots=4$.

Nun nutzen wir die Invariante: Wir wissen, dass das gesamte Spielfeld aufsummiert 4 ergibt. Da die unteren drei Felder sich bereits zu 2 aufsummieren, beträgt die Summe aller anderen Felder auch 2. Die Invariante besagt, dass nach beliebigen Operationen immer Felder mit Summe 2 belegt sind. Angenommen, die unteren Felder wären irgendwann leer, dann würden wir alle anderen unendlich vielen Felder belegen. Um allerdings soviele Felder zu belegen, müssten wir tatsächlich unendlich lange spielen und würden diesen Zustand nie erreichen.