Diskrete Modellierung

Wintersemester 2023/24

Prof. Dr. Martin Hoefer
Tim Koglin, Conrad Schecker

GOETHE UNIVERSITÄT FRANKFURT AM MAIN
Institut für Informatik
Algorithmen und Komplexität

Übungsblatt 6

Ausgabe: 04.12.2023
Abgabe: 11.12.2023, 23:55 Uhr

Aufgabe 6.1. *Weihnachtsmarkt* $(4 + 4 + 4 = 12$ Punkte)

Ein Standbetreiber des Weihnachtsmarktes möchte die Anzahl der Kunden a_n für jeden Tag $n \in \mathbb{N}_{>0}$ nach Eröffnung des Marktes abschätzen. Da zufriedene Kunden für gewöhnlich andere Kunden von der Qualität der Produkte überzeugen, hängt a_n unter anderem von der Anzahl der Kunden der vorherigen Tage ab. Es werden verschiedene Annahmen getroffen:

a) Angenommen, es gelte $a_{n+1} = 2a_n + 1$ für alle $n \geq 1$, sowie $a_1 = 7$.

b) Angenommen, es gelte $a_{n+2} = a_n + 4$ für alle $n \geq 1$, sowie $a_1 = 3$, $a_2 = 5$.

c) Angenommen, es gelte $a_n = (a_{n-1})^2$ für alle $n \geq 2$, sowie $a_1 = 2$.

Geben Sie jeweils einen geschlossenen, nicht-rekursiven Ausdruck für a_n an und beweisen Sie seine Korrektheit per vollständiger Induktion.

Aufgabe 6.2. *Bobs opulente Bäckerei* $(2 + 1 + 6 + 2 = 11$ Punkte)

Bobs opulente Bäckerei hat einen treuen Kundenstamm aus insgesamt $m \in \mathbb{N}_{>0}$ Personen, die zur Weihnachtszeit ein kleines Geschenk erhalten sollen. Bob stellt $n \in \mathbb{N}_{>0}$ verschiedene, originelle Weihnachtsspezialitäten her - inklusive der neusten Kreation. Diese sollen auf m äußerlich ununterscheidbare Geschenkboxen verteilt werden. Dabei ist Neid möglichst zu vermeiden, während sich Bob außerdem eine möglichst breite Rückmeldung über die Spezialitäten erhofft. Daher gilt für eine gültige Aufteilung: Jede Geschenkbox enthält genau eine Weihnachtsspezialität und jede Weihnachtsspezialität wird höchstens einmal vergeben. Für gegebene n, m bezeichne $T(n, m)$ die Anzahl der verschiedenen gültigen Aufteilungen, wobei symmetrische Aufteilungen (aufgrund der Ununterscheidbarkeit der Geschenkboxen) nicht mehrfach gezählt werden.

Lösen Sie die folgenden Teilaufgaben jeweils mit kurzer Begründung:

a) Geben Sie alle gültigen Aufteilungen für den Fall $n = 5$, $m = 2$ sowie deren Anzahl $T(5, 2)$ an.

b) Zunächst überlegt sich Bob, ob die neuste Kreation überhaupt verschenkt werden sollte oder nicht. Produktchefin Alice merkt an, dass sich das Problem mit der Aufteilung vereinfacht, wenn die neuste Kreation ebenfalls in eine Geschenkbox gepackt werden würde. In diesem Fall würden nämlich nur noch $m - 1$ leere Geschenkboxen übrig bleiben, die bepackt werden müssten. Dafür stehen noch $n - 1$ verschiedene Weihnachtsspezialitäten zur Verfügung.

   Angenommen, die neuste Kreation wird *nicht* in eine Geschenkbox gepackt: Wie lautet dann die Anzahl möglicher Aufteilungen?

c) Entwerfen Sie mithilfe der Beobachtungen aus der vorherigen Teilaufgabe eine rekursive Berechnungsmethode für $T(n, m)$ für beliebige $n, m \in \mathbb{N}_{>0}$. Geben Sie hierfür Basisfälle und Rekursionsfälle an und begründen Sie diese jeweils.

d) Berechnen Sie $T(6, 3)$ mithilfe der rekursiven Berechnungsmethode. Ein nachvollziehbarer Rechenweg ist als Begründung ausreichend. Sie dürfen das Ergebnis von $T(5, 2)$ aus der ersten Teilaufgabe einsetzen.

Bitte wenden!

Aufgabe 6.3. *Korrektheit eines rekursiven Programms* (10 Punkte)

Der folgende rekursive Algorithmus soll für eine beliebige Menge $A = \{a_1, \dots, a_m\}$ mit $m \in \mathbb{N}_{>0}$ Elementen die Menge aller möglichen Anordnungen der Elemente von A ausgeben. Eine Anordnung der Elemente von A ist ein m-Tupel, das genau diese Elemente (in irgendeiner Reihenfolge) enthält. Sie dürfen annehmen, dass es $m!$ paarweise verschiedene Anordnungen der Elemente von A gibt.

```
function perm({a_1, ..., a_m}):
    if m = 1 then
        return {(a_1)};
    else
        M := ∅;
        S := perm({a_1, ..., a_{m-1}});                    /* Alle Anordnungen ohne a_m. */
        foreach (x_1, ..., x_{m-1}) ∈ S do
            t_1 := (a_m, x_1, ..., x_{m-1}) ;              /* Füge a_m an Stelle 1 ein. */
            t_m := (x_1, ..., x_{m-1}, a_m) ;              /* Füge a_m an Stelle m ein. */
            M := M ∪ {t_1, t_m};
            for i = 2, ..., m − 1 do
                t := (x_1, ..., x_{i-1}, a_m, x_i ..., x_{m-1}) ;   /* Füge a_m an Stelle i ein. */
                M := M ∪ {t};
            end
        end
        return M;
    end
end
```

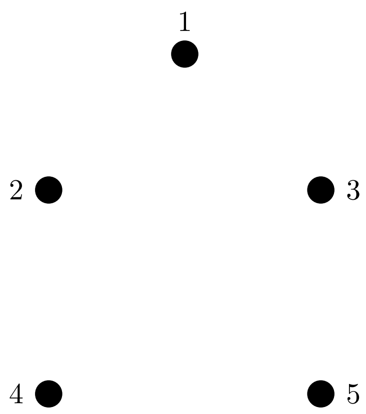
Zeigen Sie mittels vollständiger Induktion, dass der Algorithmus korrekt arbeitet, also für jede Eingabemenge $\{a_1, \dots, a_m\}$ die Menge aller Anordnungen von $a_1, \dots, a_m$ ausgibt.

Hinweis: Argumentieren Sie z.B. über die Anzahl der paarweise verschiedenen Anordnungen.

Aufgabe 6.4. *Aus gegebenem Anlass* (2 Punkte)

Ein *Eulerweg* ist ein Weg in einem Graphen, der jede Kante genau einmal durchläuft. Ein ungerichteter zusammenhängender Graph enthält genau dann einen Eulerweg, wenn zwei oder keiner seiner Knoten von ungeradem Grad sind. Hat kein Knoten ungeraden Grad, handelt es sich bei dem Eulerweg um einen *Eulerkreis*.

Gegeben sei ein ungerichteter Graph $G = (V, E)$ mit $|V| = 5$ Knoten. Zeichnen Sie in die unten gegebene Vorlage eine Kantenmenge E aus $|E| = 8$ Kanten ein, so dass der resultierende Graph G einen Eulerweg, jedoch keinen Eulerkreis enthält. Geben Sie einen möglichen Eulerweg explizit an.



Bitte wenden!

Aufgabe 6.5. *Ausfallsichere Topologien* (3 + (3 + 3 + 6) = 15 Punkte)

Für die Ausfallsicherheit von Netzwerkinfrastruktur spielt die Netzwerktopologie eine entscheidende Rolle. Um eine Menge von Rechnern ausfallsicher(er) über Leitungen zu vernetzen, sollte das Netzwerk zumindest die folgenden Eigenschaften aufweisen:

(1) Jeder Rechner ist über mindestens zwei Leitungen an andere Rechner angeschlossen.

(2) Jeder Rechner kann jeden anderen Rechner über einen Leitungsweg erreichen.

(3) Auch wenn genau ein Rechner mitsamt all seinen angeschlossenen Leitungen ausfällt, müssen alle anderen Rechner noch über mindestens einen Leitungsweg miteinander verbunden sein.

Dabei wird angenommen, dass auf jeder Leitung Daten in beide Richtungen gesendet werden können, sofern die Rechner an den beiden Enden der Leitung nicht ausgefallen sind.

Ein solches Netzwerk kann als ungerichteter Graph dargestellt werden: Jeder Knoten repräsentiert einen Rechner und jede Kante repräsentiert eine Leitung.

a) Formulieren Sie die Eigenschaften (1), (2) und (3) jeweils als graphentheoretische Aussage mit den Definitionen aus der Vorlesung. Eine Begründung ist nicht erforderlich.

b) Sei $n \in \mathbb{N}$ mit $n \geq 3$ und $V := \{1, \ldots, n\}$.

   Geben Sie für jede der Eigenschaften (1), (2) und (3) an, für welche Werte n diese in den durch die Graphen G_1, G_2 und G_3 beschrieben Topologien erfüllt ist und für welche nicht.

   i) $G_1 = (V, E_1)$ mit $E_1 = \big\{\{1, i\} : i \in V,\ i \neq 1\big\}$

   ii) $G_2 = (V, E_2)$ mit $E_2 = E_1 \cup \{\{2, n\}\} \cup \big\{\{i, i+1\} \ :\ i \in V,\ 1 < i < n\big\}$

   iii) $G_3 = (V, E_3)$ mit $E_3 = E_3^a \cup E_3^b \cup E_3^c \cup E_3^d \cup E_3^e$, wobei

$$\begin{aligned}
E_3^a &= \{\{i, i-1\} : i = 2 + 3j, j \in \mathbb{N}, i \leq n\},\\
E_3^b &= \{\{i, i-1\} : i = 3j, j \in \mathbb{N}_{>0}, i \leq n\},\\
E_3^c &= \{\{i, i-2\} : i = 3j, j \in \mathbb{N}_{>0}, i \leq n\},\\
E_3^d &= \{\{i, i-1\} : i = 4 + 3j, j \in \mathbb{N}, i \leq n\},\\
E_3^e &= \{\{i, i-2\} : i = 4 + 3j, j \in \mathbb{N}, i \leq n\}.
\end{aligned}$$