Diskrete und Geometrische Algorithmen

4. Übung am 16.11.2020

Richard Weiss   Florian Schager   Christian Sallinger   Fabian Zehetgruber
Paul Winkler   Christian Göth

Aufgabe 19. Lösen Sie die Rekursion $na_n = (n+1)a_{n-1} + 2n$ mit $a_0 = 0$.

Lösung. Wir dividieren auf beiden Seiten durch n und erhalten die Rekursionsgleichung

$$a_n = \frac{n+1}{n}a_{n-1} + 2$$

Dies ist eine aus der Vorlesung, siehe dazu Satz 4.1 im Skriptum, bekannte Form mit $b_0 = 0, b_n = 2, n \geq 1, c_n = \frac{n+1}{n}$. Die Lösung ist auch aus der Vorlesung bekannt und hat die Form

$$a_n = \sum_{i=0}^{n} b_i \prod_{j=i+1}^{n} c_j = 2\sum_{i=1}^{n}\prod_{j=i+1}^{n}\frac{j+1}{j} = 2\sum_{i=1}^{n}\frac{n+1}{i+1} = 2(n+1)\sum_{i=2}^{n+1}\frac{1}{i}$$

Eine Lösung der Rekursionsgleichung ist. Mit Induktion nach n erhalten wir für festes i

$$\prod_{j=i+1}^{i} = 1 = \frac{i+1}{i+1}, \quad \prod_{j=i+1}^{n}\frac{j+1}{j} = \frac{n+1}{n}\frac{n}{i+1} = \frac{n+1}{i+1}$$

Machen wir noch die Probe.

$$\begin{aligned}
a_0 &= 0,\\
2n(n+1)\sum_{i=2}^{n+1}\frac{1}{i} &= na_n \overset{!}{=} (n+1)a_{n-1} + 2n = (n+1)2n\sum_{i=2}^{n}\frac{1}{i} + 2n\\
&= 2n(n+1)\sum_{i=2}^{n}\frac{1}{i} + 2n(n+1)\frac{1}{n+1} = \quad 2n(n+1)\sum_{i=2}^{n+1}\frac{1}{i}
\end{aligned}$$

□

Aufgabe 20. Lösen Sie folgendes System von Rekursionen

$$a_{n+1} = 2a_n + b_n, \quad b_{n+1} = 4a_n - b_n, \quad \text{für } n \geq 1$$

mit den Anfangsbedingungen $a_0 = 1, b_0 = 0$ indem Sie das System in eine äquivalente Rekursion 2. Ordnung umschreiben.

Lösung. Wir formen zuerst die erste Rekursion um und setzen in die zweite ein

$$\begin{aligned}
&a_n = 2a_{n-1} + b_{n-1} \Leftrightarrow b_{n-1} = a_n - 2a_{n-1}\\
\Rightarrow\ &b_n = 4a_{n-1} - b_{n-1} = 4a_{n-1} - (a_n - 2a_{n-1}) = 6a_{n-1} - a_n\\
\Rightarrow\ &b_{n-1} = 6a_{n-2} - a_{n-1}
\end{aligned}$$

Setzen wir nun b_{n-1} in die Gleichung aus der ersten Zeile ein erhalten wir also

$$a_n = 2a_{n-1} + b_{n-1} = 2a_{n-1} + 6a_{n-2} - a_{n-1} = a_{n-1} + 6a_{n-2}, \quad \text{für } n \geq 2$$

Mit den Anfangsbedingungen $a_0 = 1, a_1 = 2a_0 + b_0 = 2$. Diese Rekursion 2. Ordnung lösen wir nun mithilfe von Satz 4.2, das charakteristische Polynom unserer Rekursionsgleichung lautet:

$$\chi(z) = z^2 - z - 6 = (z-3)(z+2)$$

Allgemeine Lösungen unserer Rekursion haben also die Form:

$$a_n = c_0(-2)^n + c_1 3^n$$

Jetzt lösen wir noch für die Anfangsbedingungen:

$$\begin{aligned} 1 &= a_0 \stackrel{!}{=} c_0 + c_1 \\ 2 &= a_1 \stackrel{!}{=} -2c_0 + 3c_1 = 2(c_1 - 1) + 3c_1 = 5c_1 - 2 \iff c_1 = \frac{4}{5} \iff c_0 = \frac{1}{5}. \end{aligned}$$

Die konkrete Lösung lautet also für $n \geq 1$

$$\begin{aligned} a_n &= \frac{1}{5}(-2)^n + \frac{4}{5}3^n \\ b_n &= 6a_{n-1} - a_n = \frac{6}{5}(-2)^{n-1} + \frac{24}{5}3^{n-1} - \frac{1}{5}(-2)^n - \frac{4}{5}3^n \\ &= \frac{8}{5}(-2)^{n-1} + \frac{12}{5}3^{n-1} = \frac{4}{5}3^n - \frac{4}{5}(-2)^n \end{aligned}$$

□

Aufgabe 21. Gegeben sei die durch folgenden Algorithmus definierte Funktion

$F(n, a, b, c)$:
if $n = 0$ **then**
  return 1
end if
if $n = 1$ **then**
  return 2
end if
$A = F(n-2, b, c, a)$
$B = F(n-2, c, a, b)$
$C = F(n-2, b, a, c)$
if $F(n-1, A, B, C) \equiv 0 \pmod 3$ **then**
  return $F(n-1, A-1, B, C+1) + F(0, A, B, C)$
else
  return $F(n-1, B-1, A+B+C, A \cdot C) + F(0, A, B, C)$
end if

Bezeichne $a(n)$ die Anzahl der Aufrufe von $F(0, x, y, z)$ mit irgendwelchen Parametern x, y und z bei der Berechnung von $F(n, a, b, c)$. Bestimmen Sie eine Rekursionsgleichung (inklusive Anfangsbedingungen) für $(a(n))_{n \in \mathbb{N}}$ und lösen Sie diese.

Lösung. Klarerweise gilt $a(0) = 1, a(1) = 0$. Wir erhalten außerdem die Rekursion

$$a(n) = 2a(n-1) + 3a(n-2) + 1.$$

Diese inhomogene, lineare Rekursionsgleichung 2-ter Ordnung mit konstanten Koeffizienten lässt sich durch $a(n) = y(n) - \frac{d}{C-1} = y(n) - \frac{1}{4}$ lösen, wobei $y(n)$ Lösung von

$$\begin{aligned} y(0) &= a(0) + \frac{1}{4} = \frac{5}{4}, \quad y(1) = a(1) + \frac{1}{4} = \frac{1}{4}, \\ y(n) &= 2y(n-1) + 3y(n-2), \quad n \geq 2. \end{aligned}$$

Das charakteristische Polynom davon lautet

$$\chi(\lambda) = \lambda^2 - 2\lambda - 3 \implies \lambda_{1,2} = 1 \pm 2.$$

Also erhalten wir die Lösung

$$y(n) = c_0(-1)^n + c_1(3)^n.$$

Lösen wir die Konstanten nach den Anfangsbedingungen auf:

$$\begin{aligned} &\frac{5}{4} = y(0) \stackrel{!}{=} c_0 + c_1 \\ &\frac{1}{4} = y(1) \stackrel{!}{=} -c_0 + 3c_1 = 3c_1 + c_1 - \frac{5}{4} \iff c_1 = \frac{3}{8} \iff c_0 = \frac{7}{8} \end{aligned}$$

Schließlich erhalten wir für $a(n)$:

$$a(n) = y(n) - \frac{1}{4} = \frac{7}{8}(-1)^n + \frac{3}{8}3^n - \frac{1}{4}$$

□

Aufgabe 22. Die Türme von Hanoi: Gegeben seien drei Stäbe A, B, C und n verschieden große Scheiben. Anfangs seien alle Scheiben der Größe nach auf Stab A aufgereiht, die größte ganz unten. Dieser Turm von Scheiben soll nun unter folgenden Regeln von A nach B transferiert werden:

- In jedem Zug darf nur eine Scheibe bewegt werden.
- Eine größere Scheibe darf nie über einer kleineren platziert werden.

Sei a_n die minimale Anzahl der benötigten Züge. Ermitteln Sie a_n durch Aufstellen und Lösen einer Rekursion.

Lösung. In den Regeln in der Angabe ist nicht explizit verlangt, dass immer nur die oberste Scheibe bewegt werden kann. Also bewegen wir einfach die unterste zuerst und erhalten einfach $a_n = n$. :)
Ernsthafte Lösung:
Wir haben in EProg einen rekurisven Algorithmus dafür programmiert, der folgende Rekursion verwendet:

1. Verschiebe die obersten $n-1$ Scheiben von Pfosten A auf Pfosten B.
2. Verschiebe die größte Scheibe von Pfosten A auf Pfosten C.
3. Verschiebe die $n-1$ Scheiben von Pfosten B auf Pfosten C.

Also erhalten wir für a_n folgende Rekursionsgleichung:

$$a_n = 2a_{n-1} + 1, \quad n \geq 2.$$

mit der expliziten Lösung

$$a_n = \sum_{i=1}^{n} \prod_{j=i+1}^{n} 2 = 2^n - 1.$$

Warum ist die Anzahl der Züge minimal? Für $n = 0$ ergibt klarerweise unsere Formeln mit 0 Zügen sicher die minimale Anzahl an Zügen zurück. Das Weitere folgt induktiv. Beginnend mit $n+1$ Scheiben auf A müssen wir, um die größte Scheibe auf C zu bringen, zuerst alle n anderen Scheiben auf B legen. Dafür brachen wir mindestens a_n Schritte. Danach mindestens nocheinmal a_n um diese n von B auf C zu bringen. □

Aufgabe 23. In Übungsblatt 3 haben wir gesehen, wie sich die Fibonacci-Zahlen mithilfe von Potenzen der Matrix $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ effizient berechnen lassen. Man kann diese Methode adaptieren, um Zahlenfolgen, die andere Rekursionsgleichungen erfüllen, zu berechnen. Wir betrachten im Folgenden die Potenzen der Matrix

$$M = \begin{pmatrix} 3 & 2 \\ 2 & 0 \end{pmatrix}.$$

Definiere

$$M_n := M^n = \begin{pmatrix} 3 & 2 \\ 2 & 0 \end{pmatrix}^n = \begin{pmatrix} a_n & b_n \\ c_n & d_n \end{pmatrix}$$

und betrachte die Koeffizienten dieser Matrix.

a) Geben Sie eine Rekursion für die Zahlenfolge $(a_n)_{n \geq 0}$ an.

b) Lösen Sie diese Rekursion, um eine explizite Formel für a_n zu erhalten.

c) Geben Sie einen Algorithmus zur effizienten Berechnung von M^n an und analysieren Sie dessen Laufzeit in Abhängigkeit von n ($\mathcal{O}$-Notation genügt).

Lösung.

a)

$$\begin{pmatrix} a_n & b_n \\ c_n & d_n \end{pmatrix} = \begin{pmatrix} 3 & 2 \\ 2 & 0 \end{pmatrix} \cdot \begin{pmatrix} a_{n-1} & b_{n-1} \\ c_{n-1} & d_{n-1} \end{pmatrix} = \begin{pmatrix} 3a_{n-1} + 2c_{n-1} & 3b_{n-1} + 2d_{n-1} \\ 2a_{n-1} & 2b_{n-1} \end{pmatrix}$$

Wir erhalten die Rekursion

$$a_n = 3a_{n-1} + 2c_{n-1} = 3a_{n-1} + 4a_{n-2}.$$

b) Nullstellen des charakteristischen Polynoms:

$$\chi(\lambda) = \lambda^2 - 3\lambda - 4, \quad \lambda_{1,2} = \frac{3}{2} \pm \sqrt{\frac{9}{4} + 4} = \frac{3}{2} \pm \frac{5}{2}$$

allgemeine Lösung:

$$a_n = \alpha_0(-1)^n + \alpha_1 4^n$$

Anfangsbedingungen:

$$\begin{aligned}
a_0 &= 1 \stackrel{!}{=} \alpha_0 + \alpha_1 \\
a_1 &= 3 \stackrel{!}{=} -\alpha_0 + 4\alpha_1 = 4\alpha_1 + \alpha_1 - 1 \iff \alpha_1 = \frac{4}{5} \iff \alpha_0 = \frac{1}{5}.
\end{aligned}$$

explizite Lösung:

$$\begin{aligned}
a_n &= \frac{1}{5}(-1)^n + \frac{4}{5}4^n \\
c_n &= 2a_{n-1} = \frac{2}{5}(-1)^{n-1} + \frac{8}{5}4^{n-1} = -\frac{2}{5}(-1)^n + \frac{2}{5}4^n
\end{aligned}$$

c) Wir erkennen aus der Darstellung in a), dass wir b_n, d_n durch die selbe Rekursiongleichung berechnen können mit Anfangswerten $b_0 = 0, b_1 = 2$.

$$\begin{aligned}
b_0 &= 0 \stackrel{!}{=} \beta_0 + \beta_1 \\
b_1 &= 2 \stackrel{!}{=} -\beta_0 + 4\beta_1 = 4\beta_1 + \beta_1 \iff \beta_1 = \frac{2}{5} \iff \beta_0 = -\frac{2}{5}.
\end{aligned}$$

explizite Lösung von b_n, d_n:

$$\begin{aligned}
b_n &= -\frac{2}{5}(-1)^n + \frac{2}{5}4^n \\
d_n &= 2b_{n-1} = -\frac{4}{5}(-1)^{n-1} + \frac{4}{5}4^{n-1} = \frac{4}{5}(-1)^n + \frac{1}{5}4^n
\end{aligned}$$

Also besteht der Aufwand des Algorithmus lediglich darin, die 4-er Potenzen zu berechnen, sowie eine konstante Anzahl von Additionen und Multiplikationen. Wie beim Algorithmus zur expliziten Berechnung der Fibonacci-Zahlen können wir $n = \sum_{i=0}^{k} a_i 2^i$ in Binärform darstellen und $4^n = \prod_{i=0}^{k} 4^{a_i 2^i}$ in logarithmischer Zeit berechnen, insgesamt erhalten wir also $\mathcal{O}(\log(n))$ für den Aufwand. Im folgenden Algorithmus ist $m \leq \lfloor \log(n) \rfloor$.

```
Prozedur ZahlPotenzieren(x, n)
    u := Binärkoeffizienten(n)
    m := u.Länge
    y := 1
    Für k = 0, ..., m
        Falls a_k = 1
            y := y · x
        Ende Falls
        Falls k < m
            x := x^2
        Ende Falls
    Ende Für
    Antworte y
Ende Prozedur
```

□

Aufgabe 24.

a) Lösen Sie die Rekursion $T(1) = 2$ und $T(n) = T(n-1) + 2$ für $n \geq 2$, indem Sie widerholt in die Rekursion einsetzen, bis Sie $T(n)$ erkennen. Verifizieren Sie ihr Ergebnis anschließend mit der Substitutionsmethode.

b) Lösen Sie die Rekursion $T(1) = 2$ und $T(n) = 3T(n-1) + 2$ für $n \geq 2$, indem Sie widerholt in die Rekursion einsetzen, bis Sie $T(n)$ erkennen. Verifizieren Sie ihr Ergebnis anschließend mit der Substitutionsmethode.

Lösung.

a) Setzen wir also zuerst ein paar mal ein:

$$\begin{aligned} T(1) &= 2 \\ T(2) &= 4 \\ T(3) &= 6 \end{aligned}$$

Die Vermutung ist, dass $T(n) = 2n$. Mit Induktion lässt sich dies leicht bestätigen:
IA($n = 2$):

$$T(2) = 4 = 2 \cdot 2$$

IV: $T(n) = 2n$

IS:$n \mapsto n+1$

$$T(n+1) = T(n) + 2 \overset{IV}{=} 2n + 2 = 2(n+1)$$

b) Einsetzen in die Rekursion liefert

$$\begin{aligned} T(1) &= 2 \\ T(2) &= 8 \\ T(3) &= 26 \\ T(4) &= 80 \end{aligned}$$

Die Vermutung ist, dass $T(n) = 3^n - 1$. Wiederum zeigen wir das mit Induktion:
IA($n = 2$):

$$T(2) = 8 = 3^2 - 1$$

IV: $T(n) = 3^n - 1$

IS:$n \mapsto n+1$

$$T(n+1) = 3T(n) + 2 \overset{IV}{=} 3(3^n - 1) + 2 = 3^{n+1} - 1$$

$\square$