Kapitel 3

Komplexität von Algorithmen

In diesem Kapitel führen wir Rekurrenz-Gleichungen[1] ein und zeigen, wie diese in einfachen Fällen gelöst werden können. Außerdem stellen wir die O-Notation vor. Diese beiden Begriffe benötigen wir, um die Laufzeit von Algorithmen analysieren zu können. Die Algorithmen selber stehen in diesem Kapitel noch im Hintergrund.

3.1 Die Fibonacci-Zahlen

Wir wollen *Rekurrenz-Gleichungen* an Hand eines eher spielerischen Beispiels einführen. Dazu betrachten wir eine Kaninchen-Farm, für die wir einen Geschäftsplan erstellen wollen. Wir beschäftigen uns hier nur mit der Frage, wie sich eine Kaninchen-Population entwickelt. Wir gehen dabei von folgenden vereinfachenden Annahmen aus:

1. Jedes Kaninchen-Paar bringt jeden Monat ein neues Kaninchen-Paar zur Welt.
2. Kaninchen haben nach zwei Monaten zum ersten Mal Junge.
3. Kaninchen leben ewig.

Wir nehmen nun an, wir hätten ein neugeborenes Kaninchen-Paar und stellen uns die Frage, wie viele Kaninchen-Paare wir nach n Monaten haben. Bezeichnen wir die Zahl der Kaninchen-Paare nach n Monaten mit $k(n)$, so gilt:

1. $k(0) = 1$

   Wir starten mit einem neugeborenem Kaninchen-Paar.

2. $k(1) = 1$

   Kaninchen bekommen das erste Mal nach zwei Monaten Junge, also hat sich die Zahl der Kaninchen-Paare nach einem Monat noch nicht verändert.

3. $k(2) = 1 + 1$

   Nach zwei Monaten bekommt unser Kaninchen-Paar zum ersten Mal Junge.

4. Allgemein gilt nach $n + 2$ Monaten:

   $$k(n + 2) = k(n + 1) + k(n)$$

   Alle Kaninchen-Paare, die zum Zeitpunkt n schon da sind, bekommen zum Zeitpunkt $n + 2$ Junge. Dies erklärt den Term $k(n)$. Da wir zur Vereinfachung unserer Rechnung von genetisch manipulierten unsterblichen Kaninchen ausgehen, sind alle Kaninchen, die zum Zeitpunkt $n + 1$ vorhanden sind, auch noch zum Zeitpunkt $n + 2$ vorhanden. Dies erklärt den Term $k(n + 1)$.

[1] Rekurrenz-Gleichungen werden in der Literatur auch als *Rekursions-Gleichungen* bezeichnet.

Die Folge der Zahlen $\big(k(n)\big)_{n\in\mathbb{N}}$ heißt Folge der *Fibonacci-Zahlen*. Das *Java*-Programm in Abbildung 3.1 auf Seite 14 berechnet diese Zahlen.

```java
public class Fibonacci
{
    public static void main(String[] args) {
        for (int i = 0; i < 100; ++i) {
            int n = fibonacci(i);
            System.out.printf("fibonacci(%d) = %d\n", i, n);
        }
    }

    public static int fibonacci(int n) {
        if (n == 0) return 1;
        if (n == 1) return 1;
        return fibonacci(n - 1) + fibonacci(n - 2);
    }
}
```

Abbildung 3.1: Ein *Java*-Programm zur Berechnung der Fibonacci-Zahlen.

Wenn wir dieses Programm laufen lassen, stellen wir fest, dass die Laufzeiten mit wachsendem Parameter n sehr schnell anwachsen. Um dieses Phänomen zu analysieren, untersuchen wir exemplarisch, wie viele Additionen bei der Berechnung von `fibonacci`(n) für ein gegebenes $n \in \mathbb{N}$ benötigt werden. Bezeichnen wir diese Zahl mit a_n, so finden wir:

1. $a_0 = 0$.
2. $a_1 = 0$.
3. $n \geq 2 \rightarrow a_n = a_{n-1} + a_{n-2} + 1$,

   denn in den rekursiven Aufrufen `fibonacci`$(n-1)$ und `fibonacci`$(n-2)$ haben wir a_{n-1} bzw. a_{n-2} Additionen und dazu kommt noch die Addition der Werte `fibonacci`$(n-1)$ und `fibonacci`$(n-2)$.

Wir setzen in der Gleichung $a_n = 1 + a_{n-1} + a_{n-2}$ für n den Wert $i+2$ ein und haben dann

$$a_{i+2} = a_{i+1} + a_i + 1 \tag{1}$$

Eine solche Gleichung nennen wir eine *lineare inhomogene Rekurrenz-Gleichung*. Die dieser Gleichung zugeordnete *homogene Rekurrenz-Gleichung* lautet

$$a_{i+2} = a_{i+1} + a_i \tag{2}$$

Wir lösen diese Gleichung mit folgendem Ansatz:

$$a_i = \lambda^i.$$

Einsetzen dieses Ansatzes in $(*)$ führt auf die Gleichung

$$\lambda^{i+2} = \lambda^{i+1} + \lambda^i.$$

Wenn wir beide Seiten dieser Gleichung durch λ^i dividieren, erhalten wir die quadratische Gleichung

$$\lambda^2 = \lambda + 1,$$

die wir mit Hilfe einer quadratischen Ergänzung lösen:

$$
\begin{array}{lcll}
\lambda^2 & = & \lambda + 1 & \mid -\lambda \\
\lambda^2 - 2 \cdot \frac{1}{2}\lambda & = & 1 & \mid +\frac{1}{4} \\
\lambda^2 - 2 \cdot \frac{1}{2}\lambda + \left(\frac{1}{2}\right)^2 & = & \frac{5}{4} & \\
\left(\lambda - \frac{1}{2}\right)^2 & = & \frac{5}{4} & \mid \sqrt{\ } \\
\lambda - \frac{1}{2} & = & \pm\frac{\sqrt{5}}{2} & \mid + \frac{1}{2} \\
\lambda_{1/2} & = & \frac{1}{2}(1 \pm \sqrt{5}) &
\end{array}
$$

Wir bemerken, dass jede Linear-Kombination der Form

$$a_n = \alpha \cdot \lambda_1^n + \beta \cdot \lambda_2^n$$

eine Lösung der homogenen Rekurrenz-Gleichung (2) ist. Wir bemerken weiter, dass für die Lösungen λ_1 und λ_2 folgende Identitäten gelten:

$$\lambda_1 - \lambda_2 = \sqrt{5} \quad \text{und} \quad \lambda_1 + \lambda_2 = 1. \tag{3}$$

Aus der letzen Gleichung folgt dann sofort

$$1 - \lambda_1 = \lambda_2 \quad \text{und} \quad 1 - \lambda_2 = \lambda_1 \tag{4}$$

Um nun die ursprüngliche Rekurrenz-Gleichung (1) zu lösen, machen wir den Ansatz $a_i = c$. Setzen wir diesen Ansatz in der Gleichung (1) ein, so erhalten wir die Gleichung

$$c = c + c + 1,$$

die die Lösung $c = -1$ hat. Diese Lösung bezeichnen wir als eine *spezielle Lösumg*. Die *allgemeine Lösung* der Rekurrenz-Gleichung (1) ergibt sich als Summe aus der Lösung der homogenen Rekurrenz-Gleichung und der speziellen Lösung und lautet daher

$$a_i = \alpha \cdot \lambda_1^i + \beta \cdot \lambda_2^i - 1$$

mit $\lambda_1 = \frac{1}{2}(1 + \sqrt{5})$ und $\lambda_2 = \frac{1}{2}(1 - \sqrt{5})$. Die Koeffizienten α und β sind jetzt so zu bestimmen, dass die Anfangs-Bedingungen $a_0 = 0$ und $a_1 = 0$ erfüllt sind. Das führt auf folgendes lineares Gleichungs-System:

$$
\begin{array}{rcl}
0 & = & \alpha \cdot \lambda_1^0 + \beta \cdot \lambda_2^0 - 1 \\
0 & = & \alpha \cdot \lambda_1^1 + \beta \cdot \lambda_2^1 - 1
\end{array}
$$

Addieren wir bei beiden Gleichungen 1 und vereinfachen für $i = 1, 2$ die Potenzen λ_i^0 zu 1 und λ_i^1 zu λ_i, so erhalten wir:

$$
\begin{array}{rcl}
1 & = & \alpha + \beta \\
1 & = & \alpha \cdot \lambda_1 + \beta \cdot \lambda_2
\end{array}
$$

Die erste dieser beiden Gleichungen liefert die Beziehung $\alpha = 1 - \beta$. Setzen wir dies für α in der zweiten Gleichung ein, so erhalten wir

$$
\begin{array}{lrcl}
 & 1 & = & (1 - \beta) \cdot \lambda_1 + \beta \cdot \lambda_2 \\
\Leftrightarrow & 1 & = & \lambda_1 + \beta \cdot \big(\lambda_2 - \lambda_1\big) \\
\Leftrightarrow & 1 - \lambda_1 & = & \beta \cdot \big(\lambda_2 - \lambda_1\big) \\
\Leftrightarrow & \dfrac{1 - \lambda_1}{\lambda_2 - \lambda_1} & = & \beta
\end{array}
$$

Wegen $\alpha = 1 - \beta$ finden wir dann

$$\alpha = -\frac{1 - \lambda_2}{\lambda_2 - \lambda_1}.$$

Verwenden wir hier die Gleichungen (3) und (4), so finden wir

$$\alpha = \frac{\lambda_1}{\sqrt{5}} \quad \text{und} \quad \beta = -\frac{\lambda_2}{\sqrt{5}}.$$

Damit können wir die Folge $(a_i)_i$ explizit angeben:

$$a_i = \frac{1}{\sqrt{5}} \cdot \left(\lambda_1^{i+1} - \lambda_2^{i+1}\right) - 1$$

Wegen $\lambda_1 \approx 1.61803$ und $\lambda_2 \approx -0.61803$ dominiert der erste Term der Summe und die Zahl der Additionen wächst exponentiell mit dem Faktor λ_1 an. Dies erklärt das starke Anwachsen der Rechenzeit.

Bemerkung: Die Zahl λ_1 wird auch als *goldener Schnitt* bezeichnet und spielt sowohl in der Geometrie als auch in der Kunst eine Rolle.
Die Ursache der Ineffezienz der Berechnung der Fibonacci-Zahlen ist leicht zu sehen: Berechnen wir den Wert `fibonacci(5)` mit dem Programm aus Abbildung 3.1, so müssen wir `fibonacci(4)` und `fibonacci(3)` berechnen. Die Berechnung von `fibonacci(4)` erfordert ihrerseits die Berechnung von `fibonacci(3)` und `fibonacci(2)`. Dann berechnen wir `fibonacci(3)` aber zweimal! Abbildung 3.2 zeigt den sogenannten *Rekursions-Baum* für den Aufruf von *fibonacci*(5), der den oben dargestellten Zusammenhang graphisch verdeutlicht.

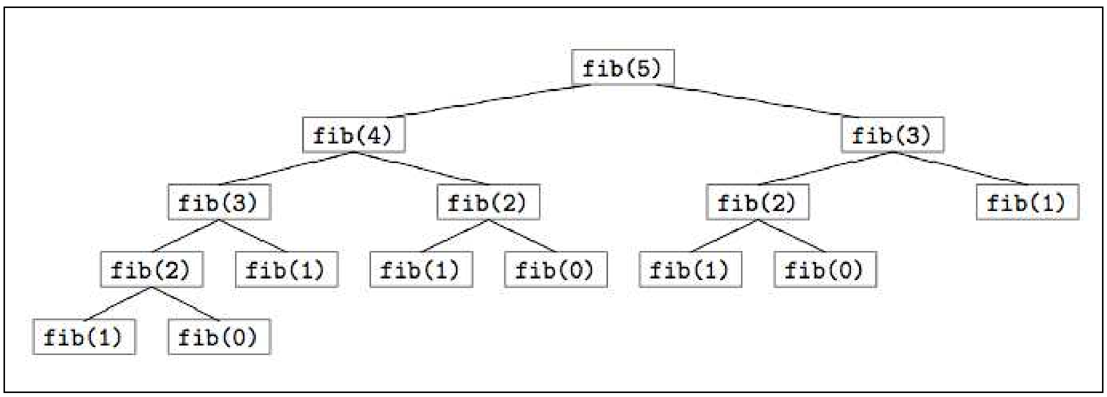

Abbildung 3.2: Rekursions-Baum für die Berechnung von *fibonacci*(5).

Wir können eine effizientere Berechnung der Fibonacci-Zahlen implementieren, indem wir uns die berechneten Werte merken. Dazu können wir in *Java* ein Feld benutzen. Dies führt zu dem in Abbildung 3.3 auf Seite 17 angegebenen Programm. Da die Werte der Funktion `fibonacci`() exponentiell wachsen, reichen 32-Bit-Zahlen nicht aus, um diese Werte darzustellen. Wir verwenden daher die Klasse `BigInteger`, mit der sich ganze Zahlen beliebiger Größe darstellen lassen. Da Felder in *Java* genau wie in `C` mit 0 beginnend indiziert werden, hat ein Feld, dessen oberster Index n ist, insgesamt $n+1$ Elemente. Wir legen daher in Zeile 19 ein Feld von $n+1$ Elementen an.

3.2 Lineare Rekurrenz-Gleichung

Wir waren bei der Analyse der Komplexität des ersten Programms zur Berechnung der Fibonacci-Zahlen auf die Gleichung

$$a_{i+2} = a_{i+1} + a_i + 1 \quad \text{für alle } i \in \mathbb{N}$$

gestoßen. Gleichungen dieser Form treten bei der Analyse der Komplexität rekursiver Programme häufig auf. Wir wollen uns daher in diesem Abschnitt näher mit solchen Gleichungen beschäftigen.

Definition 6 (Lineare homogene Rekurrenz-Gleichung)
Die lineare homogene Rekurrenz-Gleichung der Ordnung k mit konstanten Koeffizienten *hat die Form*

$$a_{n+k} = c_{k-1} \cdot a_{n+k-1} + c_{k-2} \cdot a_{n+k-2} + \cdots + c_1 \cdot a_{n+1} + c_0 \cdot a_n \quad \textit{für alle } n \in \mathbb{N}. \qquad (3.1)$$

In Summen-Schreibweise kann diese Gleichung kompakter als

```
import java.math.*;

public class FibonacciBig
{
    public static void main(String[] args)
    {
        for (int i = 0; i < 100; ++i) {
            BigInteger n = fibonacci(i);
            System.out.println("fib(" + i + ") = " + n);
        }
    }

    public static BigInteger fibonacci(int n)
    {
        if (n <= 2) {
            return BigInteger.valueOf(1);
        }
        BigInteger[] mem = new BigInteger[n+1];
        mem[0] = BigInteger.valueOf(1);  // fibonacci(0) = 1
        mem[1] = BigInteger.valueOf(1);  // fibonacci(1) = 1
        for (int i = 0; i < n - 1; ++i) {
            mem[i + 2] = mem[i].add(mem[i + 1]);
        }
        return mem[n];
    }
}
```

Abbildung 3.3: Berechnung der Fibonacci-Zahlen mit Speicherung der Zwischenwerte.

$$a_{n+k} = \sum_{i=0}^{k-1} c_i \cdot a_{n+i} \qquad \textit{für alle } n \in \mathbb{N}$$

geschreiben werden. Zusätzlich werden Anfangs-Bedingungen

$$a_0 = d_0, \cdots, a_{k-1} = d_{k-1}$$

für die Folge $\bigl(a_n\bigr)_{n\in\mathbb{N}}$ *vorgegeben.* □

Durch eine lineare homogene Rekurrenz-Gleichung wird die Folge $(a_n)_{n\in\mathbb{N}}$ eindeutig bestimmt: Die Werte a_n für $n < k$ sind durch die Anfangs-Bedingungen gegeben und alle weiteren Werte können dann durch die Rekurrenz-Gleichung (3.1) bestimmt werden. Noch etwas zur Nomenklatur:

1. Die Rekurrenz-Gleichung (3.1) heißt *linear*, weil die Glieder der Folge $(a_n)_n$ nur linear in der Gleichung (3.1) auftreten. Ein Beispiel für eine Rekurrenz-Gleichung, die nicht linear ist, wäre

   $$a_{n+1} = a_n^2 \qquad \text{für alle } n \in \mathbb{N}.$$

   Nicht-lineare Rekurrenz-Gleichungen sind nur in Spezialfällen geschlossen lösbar.

2. Die Rekurrenz-Gleichung (3.1) heißt *homogen*, weil auf der rechten Seite dieser Gleichung kein konstantes Glied mehr auftritt. Ein Beispiel für eine Gleichung, die nicht homogen ist (wir sprechen auch von *inhomogenen* Rekurrenz-Gleichungen), wäre

   $$a_{n+2} = a_{n+1} + a_n + 1 \qquad \text{für alle } n \in \mathbb{N}.$$

   Mit inhomogenen Rekurrenz-Gleichungen werden wir uns später noch beschäftigen.

3. Die Rekurrenz-Gleichung (3.1) hat *konstante Koeffizienten*, weil die Werte c_i Konstanten

sind, die nicht von dem Index n abhängen. Ein Beispiel für eine Rekurrenz-Gleichung, die keine konstanten Koeffizienten hat, ist

$$a_{n+1} = n \cdot a_n \qquad \text{für alle } n \in \mathbb{N}.$$

Solche Rekurrenz-Gleichungen können in vielen Fällen auf Rekurrenz-Gleichungen mit konstanten Koeffizienten zurück geführt werden. Wir werden das später noch im Detail besprechen.

Wie lösen wir eine lineare homogene Rekurrenz-Gleichung? Wir versuchen zunächst den Ansatz

$$a_n = \lambda^n \qquad \text{für alle } n \in \mathbb{N}.$$

Einsetzen dieses Ansatzes in (3.1) führt auf die Gleichung

$$\lambda^{n+k} = \sum_{i=0}^{k-1} c_i \cdot \lambda^{n+i}$$

Dividieren wir diese Gleichung durch λ^n, so haben wir:

$$\lambda^{k} = \sum_{i=0}^{k-1} c_i \cdot \lambda^{i}$$

Das Polynom

$$\chi(x) = x^k - \sum_{i=0}^{k-1} c_i \cdot x^{i}$$

heißt *charakteristisches Polynom* der Rekurrenz-Gleichung (3.1). Wir betrachten zunächst den Fall, dass das charakteristische Polynom k verschiedene Nullstellen hat. In diesem Fall sagen, dass die Rekurrenz-Gleichung (3.1) *nicht entartet* ist. Bezeichnen wir diese Nullstellen mit

$$\lambda_1,\, \lambda_2,\, \cdots,\, \lambda_k,$$

so gilt für alle $j = 1, \cdots, k$

$$\lambda_j^{n+k} = \sum_{i=0}^{k-1} c_i \cdot \lambda_j^{n+i}.$$

Damit ist die Folge

$$\bigl(\lambda_j^n\bigr)_{n\in\mathbb{N}}$$

für alle $j = 1, \cdots, k$ eine Lösung der Rekurrenz-Gleichung (3.1). Außerdem ist auch jede Linear-Kombination dieser Lösungen eine Lösung von (3.1): Definieren wir die Folge a_n durch

$$a_n = \alpha_1\lambda_1^n + \cdots + \alpha_k\lambda_k^n \qquad \text{für alle } n \in \mathbb{N}$$

mit beliebigen Koeffizienten $\alpha_i \in \mathbb{R}$, so erfüllt auch die Folge $(a_n)_n$ die Gleichung (3.1). Die oben definierte Folge $(a_n)_n$ bezeichnen wir als die *allgemeine Lösung* der Rekurrenz-Gleichung (3.1): Die Koeffizienten α_1 bis α_k müssen wir nun so wählen, dass die Anfangs-Bedingungen

$$a_0 = d_0,\, \cdots,\, a_{k-1} = d_{k-1}$$

erfüllt sind. Das liefert ein lineares Gleichungs-System für die Koeffizienten $\alpha_1, \cdots, \alpha_k$:

$$\begin{array}{lcl}
d_0 & = & \lambda_1^0 \cdot \alpha_1 + \cdots + \lambda_k^0 \cdot \alpha_k \\
d_1 & = & \lambda_1^1 \cdot \alpha_1 + \cdots + \lambda_k^1 \cdot \alpha_k \\
\vdots & & \vdots \\
d_{k-1} & = & \lambda_1^{k-1} \cdot \alpha_1 + \cdots + \lambda_k^{k-1} \cdot \alpha_k
\end{array}$$

Hier sind die Werte λ_i die Nullstellen des charakteristischen Polynoms. Die Matrix V, die diesem Gleichungs-System zugeordnet ist, lautet:

$$V = \begin{pmatrix} \lambda_1^0 & \cdots & \lambda_k^0 \\ \lambda_1^1 & \cdots & \lambda_k^1 \\ \vdots & & \vdots \\ \lambda_1^{k-1} & \cdots & \lambda_k^{k-1} \end{pmatrix}$$

Diese Matrix ist in der Mathematik als *Vandermonde*'sche Matrix bekannt. Für die Determinate dieser Matrix gilt

$$\det(V) = \prod_{1 \leq i < j \leq k} (\lambda_i - \lambda_j).$$

Sind die Nullstellen λ_i für $i = 1, \cdots, k$ paarweise verschieden, so ist jeder der Faktoren $(\lambda_i - \lambda_j)$ von 0 verschieden und damit ist auch das Produkt von 0 verschieden. Daraus folgt, das das zugehörige lineare Gleichungs-System eindeutig lösbar ist. Mit der Lösung dieses Gleichungs-Systems haben wir dann die Lösung der Rekurrenz-Gleichung (3.1) gefunden.

Beispiel: Wie demonstrieren das Verfahren an einem Beispiel: Wie betrachten die Rekurrenz-Gleichung

$$a_{n+2} = a_{n+1} + a_n \quad \text{für alle } n \in \mathbb{N}$$

mit den Anfangs-Bedingungen $a_0 = 1$ und $a_1 = 1$. Die Lösung dieser Rekurrenz-Gleichung sind übrigens gerade die Fibonacci-Zahlen. Das *charakteristische Polynom* dieser Rekurrenz-Gleichung lautet:

$$\chi(x) = x^2 - x - 1.$$

Das führt auf die quadratische Gleichung

$$x^2 - x - 1 = 0$$

Wir haben eben schon gesehen, dass diese quadratische Gleichung die Lösung

$$x_{1/2} = \tfrac{1}{2}(1 \pm \sqrt{5})$$

hat. Wir definieren

$$\lambda_1 = \tfrac{1}{2}(1 + \sqrt{5}) \quad \text{und} \quad \lambda_2 = \tfrac{1}{2}(1 - \sqrt{5}).$$

Damit lautet die *allgemeine Lösung* der betrachteten Rekurrenz-Gleichung

$$a_n = \alpha_1 \cdot \lambda_1^n + \alpha_2 \cdot \lambda_2^n \quad \text{für alle } n \in \mathbb{N}.$$

Setzen wir hier die Anfangs-Bedingungen ein, so erhalten wir

$$\begin{aligned} 1 &= \lambda_1^0 \cdot \alpha_1 + \lambda_2^0 \cdot \alpha_2 \\ 1 &= \lambda_1^1 \cdot \alpha_1 + \lambda_2^1 \cdot \alpha_2 \end{aligned}$$

Dies ist ein lineares Gleichungs-System in den Unbekannten α_1 und α_2. Vereinfachung führt auf

$$\begin{aligned} 1 &= \alpha_1 + \alpha_2 \\ 1 &= \lambda_1 \cdot \alpha_1 + \lambda_2 \cdot \alpha_2 \end{aligned}$$

Die erste dieser beiden Gleichungen lösen wir nach α_2 auf und finden $\alpha_2 = 1 - \alpha_1$. Diesen Wert setzen wir in der zweiten Gleichung ein. Das führt auf

$$\begin{aligned} & 1 &= \lambda_1 \cdot \alpha_1 + \lambda_2 \cdot (1 - \alpha_1) \\ \Leftrightarrow \quad & 1 &= (\lambda_1 - \lambda_2) \cdot \alpha_1 + \lambda_2 \\ \Leftrightarrow \quad & 1 - \lambda_2 &= (\lambda_1 - \lambda_2) \cdot \alpha_1 \\ \Leftrightarrow \quad & \frac{1 - \lambda_2}{\lambda_1 - \lambda_2} &= \alpha_1 \end{aligned}$$

Setzen wir diesen Wert in der Gleichung $\alpha_2 = 1 - \alpha_1$ ein, so erhalten wir

$$\alpha_2 = 1 - \frac{1 - \lambda_2}{\lambda_1 - \lambda_2} = \frac{\lambda_1 - 1}{\lambda_1 - \lambda_2}.$$

Berücksichtigen wir die Gleichungen (1) und (2), so finden wir:

$$\alpha_1 = \frac{\lambda_1}{\sqrt{5}} \quad \text{und} \quad \alpha_2 = -\frac{\lambda_2}{\sqrt{5}}.$$

Die Lösung der Rekurrenz-Gleichung

$$a_{n+2} = a_{n+1} + a_n \quad \text{für alle } n \in \mathbb{N}$$

mit den Anfangs-Bedingungen $a_0 = 1$ und $a_1 = 1$ lautet also

$$a_n = \frac{1}{\sqrt{5}} \cdot \left(\lambda_1^{n+1} - \lambda_2^{n+1}\right) \quad \text{für alle } n \in \mathbb{N}.$$

Damit haben wir eine geschlossene Formel zur Berechnung der Fibonacci-Zahlen gefunden. Diese Formel zeigt uns, dass die Fibonacci-Zahlen selbst exponentiell anwachsen. Wir werden diese Lösung bei der Analyse des *Quick-Sort*-Algorithmus benötigen.

Aufgabe: Lösen Sie die Rekurrenz-Gleichung $a_{n+2} = \frac{3}{2} \cdot a_{n+1} - \frac{1}{2} \cdot a_n$ mit den Anfangs-Bedingungen $a_0 = 3$ und $a_1 = \frac{5}{2}$.

3.2.1 Entartete Rekurrenz-Gleichungen

Wir hatten oben zunächst den Fall betrachtet, dass das charakteristische Polynom der Rekurrenz-Gleichung (3.1) insgesamt k verschiedene Nullstellen hat. Dies muss keineswegs immer der Fall sein. Wir betrachten die Rekurrenz-Gleichung

$$a_{n+2} = 4 \cdot a_{n+1} - 4 \cdot a_n \quad \text{für alle } n \in \mathbb{N} \tag{3.2}$$

mit den Anfangs-Bedingungen $a_0 = 1$, $a_1 = 4$. Das charakteristische Polynom lautet

$$\chi(x) = x^2 - 4 \cdot x + 4 = (x-2)^2$$

und hat offensichtlich nur eine Nullstelle bei $x = 2$. Eine Lösung der Rekurrenz-Gleichung (3.2) lautet daher

$$a_n = 2^n \quad \text{für alle } n \in \mathbb{N}.$$

Eine weitere Lösung ist

$$a_n = n \cdot 2^n \quad \text{für alle } n \in \mathbb{N}.$$

Wir verifizieren dies durch Einsetzen:

$$\begin{array}{lllll}
 & (n+2) \cdot 2^{n+2} & = & 4 \cdot (n+1) \cdot 2^{n+1} - 4 \cdot n \cdot 2^n & \mid \div 2^n \\
\Leftrightarrow & (n+2) \cdot 2^2 & = & 4 \cdot (n+1) \cdot 2^1 - 4 \cdot n & \mid \div 4 \\
\Leftrightarrow & n+2 & = & (n+1) \cdot 2 - n & \\
\Leftrightarrow & n+2 & = & 2 \cdot n + 2 - n & \\
\Leftrightarrow & n+2 & = & n+2 &
\end{array}$$

Die allgemeine Lösung der Rekurrenz-Gleichung finden wir durch Linear-Kombination der beiden Lösungen:

$$a_n = \alpha \cdot 2^n + \beta \cdot n \cdot 2^n \quad \text{für alle } n \in \mathbb{N}.$$

Setzen wir hier die Anfangs-Bedingungen $a_0 = 1$ und $a_2 = 4$ ein, so erhalten wir:

$$\left\{ \begin{array}{lll} 1 & = & \alpha \cdot 2^0 + \beta \cdot 0 \cdot 2^0 \\ 4 & = & \alpha \cdot 2^1 + \beta \cdot 1 \cdot 2^1 \end{array} \right\} \quad \Leftrightarrow \quad \left\{ \begin{array}{lll} 1 & = & \alpha \\ 4 & = & \alpha \cdot 2 + \beta \cdot 2 \end{array} \right\}$$

Die Lösung lautet offenbar $\alpha = 1$ und $\beta = 1$. Damit lautet die Lösung der Rekurrenz-Gleichung (3.2) mit den Anfangs-Bedingungen $a_0 = 1$ und $a_2 = 4$

$$a_n = 2^n + n \cdot 2^n = (n+1) \cdot 2^n \quad \text{für alle } n \in \mathbb{N}.$$

Im allgemeinen nennen wir die Rekurrenz-Gleichung

$$a_{n+k} = \sum_{i=0}^{k-1} c_i \cdot a_{n+i}$$

entartet, wenn das charakteristische Polynom

$$\chi(x) = x^k - \sum_{i=0}^{k-1} c_i \cdot x^i$$

weniger als k verschiedene Nullstellen hat. Dann läßt sich folgendes zeigen: Hat das charakteristische Polynom $\chi(x)$ eine r-fache Nullstelle λ, gilt also

$$\chi(x) = (x - \lambda)^r \cdot \phi(x)$$

mit einem geeigneten Polynom $\phi(x)$, so sind die Folgen

1. $(\lambda^n)_{n \in \mathbb{N}}$

2. $(n \cdot \lambda^n)_{n\in\mathbb{N}}$
3. $(n^2 \cdot \lambda^n)_{n\in\mathbb{N}}$
4. $\vdots$
5. $(n^{r-1} \cdot \lambda^n)_{n\in\mathbb{N}}$

Lösungen der Rekurrenz-Gleichung (3.2). Durch eine geeignete Linear-Kombination dieser Lösungen zusammen mit den Lösungen, die sich aus den Nullstellen des Polynoms ϕ ergeben, läßt sich dann immer eine Lösung finden, die auch den Anfangs-Bedingungen genügt.

Aufgabe: Lösen Sie die Rekurrenz-Gleichung

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

für die Anfangs-Bedingungen $a_0 = 0$, $a_1 = 3$, $a_2 = 2$.

3.2.2 Inhomogene Rekurrenz-Gleichungen

Definition 7 (Lineare inhomogene Rekurrenz-Gleichung) *Die* lineare inhomogene Rekurrenz-Gleichung der Ordnung k mit konstanten Koeffizienten und konstanter Inhomogenität *hat die Form*

$$a_{n+k} = \sum_{i=0}^{k-1} c_i \cdot a_{n+i} + c_{-1} \tag{3.3}$$

mit den Anfangs-Bedingungen $a_0 = d_0$, $\cdots$, $a_{k-1} = d_{k-1}$. *Dabei gilt für die Koeffizienten*

$$c_i \in \mathbb{R} \quad \textit{für alle } i = -1, 0, \cdots, k-1.$$

Für die Anfangs-Bedingungen $d_0, \cdots, d_{k-1}$ *gilt ebenfalls*

$$d_i \in \mathbb{R} \quad \textit{für alle } i = 0, \cdots, k-1.$$

Die Konstante c_{-1} *bezeichnen wir als die* Inhomogenität. □

Wie läßt sich die inhomogene Rekurrenz-Gleichung (3.3) lösen? Wir zeigen zunächst, wie sich eine *spezielle Lösung* der Rekurrenz-Gleichung (3.3) finden läßt. Dazu betrachten wir das charakteristische Polynom

$$\chi(x) = x^k - \sum_{i=0}^{k-1} c_i \cdot x^i$$

und definieren die *Spur* $\texttt{sp}(\chi)$ wie folgt:

$$\texttt{sp}(\chi) := \chi(1) = 1 - \sum_{i=0}^{k-1} c_i.$$

Es können zwei Fälle auftreten, $\texttt{sp}(\chi) \neq 0$ und $\texttt{sp}(\chi) = 0$. Wir betrachten die beiden Fälle getrennt.

1. $\texttt{sp}(\chi) \neq 0$.

   Dann erhalten wir eine spezielle Lösung von (3.3) durch den Ansatz

   $$a_n = \delta \quad \text{für alle } n \in \mathbb{N}.$$

   Den Wert von δ bestimmen wir durch Einsetzen, es muß für alle $n \in \mathbb{N}$ gelten:

   $$\delta = \sum_{i=0}^{k-1} c_i \cdot \delta + c_{-1}.$$

   Daraus ergibt sich

   $$\delta \cdot \left(1 - \sum_{i=0}^{k-1} c_i\right) = c_{-1}.$$

   Das ist aber nichts anderes als

   $$\delta \cdot \texttt{sp}(\chi) = c_{-1}$$

und damit lautet eine spezielle Lösung von (3.3)

$$a_n = \delta = \frac{c_{-1}}{\texttt{sp}(\chi)}.$$

Jetzt sehen wir auch, warum die Voraussetzung $\texttt{sp}(\chi) \neq 0$ wichtig ist, denn anderfalls wäre der Quotient $\frac{c_{-1}}{\texttt{sp}(\chi)}$ undefiniert.

2. $\texttt{sp}(\chi) = 0$.

In diesem Fall versuchen wir, eine spezielle Lösung von (3.3) durch den Ansatz

$$a_n = \varepsilon \cdot n$$

zu finden. Den Wert ε erhalten wir durch Einsetzen, es muß für alle $n \in \mathbb{N}$ gelten:

$$\varepsilon \cdot (n+k) = \sum_{i=0}^{k-1} c_i \cdot \varepsilon \cdot (n+i) + c_{-1}$$

Dies formen wir wie folgt um:

$$\varepsilon \cdot n + \varepsilon \cdot k = \varepsilon \cdot n \cdot \sum_{i=0}^{k-1} c_i + \varepsilon \cdot \sum_{i=0}^{k-1} i \cdot c_i + c_{-1}$$

Aus $\texttt{sp}(\chi) = 0$ folgt $1 = \sum_{i=0}^{k-1} c_i$ und damit gilt

$$\varepsilon \cdot n = \varepsilon \cdot n \cdot \sum_{i=0}^{k-1} c_i.$$

Daher vereinfacht sich die obige Gleichung zu

$$\begin{aligned} & \varepsilon \cdot k = \varepsilon \cdot \sum_{i=0}^{k-1} i \cdot c_i + c_{-1} \\ \Leftrightarrow \quad & \varepsilon \cdot \left(k - \sum_{i=0}^{k-1} i \cdot c_i\right) = c_{-1} \\ \Leftrightarrow \quad & \varepsilon = \frac{c_{-1}}{k - \sum\limits_{i=0}^{k-1} i \cdot c_i} \end{aligned}$$

Wenn wir genau hin schauen, dann sehen wir, dass der Wert im Nenner nicht anderes ist als der Wert der Ableitung des charakteristischen Polynoms an der Stelle 1, denn es gilt:

$$\chi'(x) = \frac{d\,\chi(x)}{dx} = k \cdot x^{k-1} - \sum_{i=0}^{k-1} c_i \cdot i \cdot x^{i-1}$$

Setzen wir hier für x den Wert 1 ein, so finden wir

$$\chi'(1) = k - \sum_{i=0}^{k-1} c_i \cdot i.$$

Insgesamt haben wir damit also die folgende spezielle Lösung $(a_n)_{n\in\mathbb{N}}$ der Gleichung (3.3) gefunden:

$$a_n = \frac{c_{-1}}{\chi'(1)} \cdot n.$$

Wir haben oben zur Vereinfachung angenommen, dass dieser Wert von 0 verschieden ist, dass also das charakteristische Polynom $\chi(x)$ an der Stelle $x = 1$ keine mehrfache Nullstelle hat, denn nur dann ist ε durch die obige Gleichung wohldefiniert und wir haben eine spezielle Lösung der Rekurrenz-Gleichung (3.3) gefunden. Andernfalls können wir die Reihe nach die Ansätze $a_n = \varepsilon \cdot n^2$, $a_n = \varepsilon \cdot n^3$, $\cdots$ versuchen, denn es kann folgendes gezeigt werden: Hat das charakteristische Polynom $\chi(x)$ am Punkt $x = 1$ eine Nullstelle vom Rang r, so führt der Ansatz $a_n = \varepsilon \cdot n^r$ zu einer speziellen Lösung von (3.3).

Diese spezielle Lösung genügt i. a. noch nicht den Anfangs-Bedingungen. Eine Lösung, die auch den Anfangs-Bedingungen genügt, erhalten wir, wenn wir zu der speziellen Lösung die allgemeine Lösung der zugehörigen homogenen linearen Rekurrenz-Gleichung

$$a_{n+k} = c_{k-1} \cdot a_{n+k-1} + c_{k-2} \cdot a_{n+k-2} + \cdots + c_1 \cdot a_{n+1} + c_0 \cdot a_n$$

addieren und die Koeffizienten der allgemeinen Lösung so wählen, dass die Anfangs-Bedingungen erfüllt sind. Wir betrachten ein Beispiel: Die zu lösende Rekurrenz-Gleichung lautet

$$a_{n+2} = 3 \cdot a_{n+1} - 2 \cdot a_n - 1 \quad \text{für alle } n \in \mathbb{N}.$$

Die Anfangs-Bedingungen sind $a_0 = 1$ und $a_1 = 3$. Wir berechnen zunächst eine spezielle Lösung. Das charakteristische Polynom ist

$$\chi(x) = x^2 - 3 \cdot x + 2 = (x-1) \cdot (x-2).$$

Es gilt $\mathtt{sp}(\chi) = \chi(1) = 0$. Wir versuchen für die spezielle Lösung den Ansatz

$$a_n = \varepsilon \cdot n.$$

Einsetzen in die Rekurrenz-Gleichung liefert

$$\varepsilon \cdot (n+2) = 3 \cdot \varepsilon \cdot (n+1) - 2 \cdot \varepsilon \cdot n - 1 \quad \text{für alle } n \in \mathbb{N}.$$

Das ist äquivalent zu

$$\varepsilon \cdot (2-3) = -1$$

und daraus folgt sofort $\varepsilon = 1$. Damit lautet eine spezielle Lösung

$$a_n = n \quad \text{für alle } n \in \mathbb{N}.$$

Da die Nullstellen des charakteristischen Polynoms $\chi(x)$ bei 1 und 2 liegen, finden wir für die allgemeine Lösung

$$a_n = \alpha \cdot 1^n + \beta \cdot 2^n + n \quad \text{für alle } n \in \mathbb{N}.$$

Setzen wir hier für n die Werte 0 und 1 und für a_n die beiden Anfangs-Bedingungen ein, so erhalten wir das Gleichungs-System

$$\left\{ \begin{array}{lcl} 1 & = & \alpha \cdot 1^0 + \beta \cdot 2^0 + 0 \\ 3 & = & \alpha \cdot 1^1 + \beta \cdot 2^1 + 1 \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{lcl} 1 & = & \alpha + \beta \\ 3 & = & \alpha + 2 \cdot \beta + 1 \end{array} \right\}$$

Sie können leicht nachrechnen, dass dieses Gleichungs-System die Lösung $\alpha = 0$ und $\beta = 1$ hat. Damit lautet die Lösung der Rekurrenz-Gleichung

$$a_n = 2^n + n \quad \text{für alle } n \in \mathbb{N}.$$

Aufgabe: Lösen Sie die inhomogene Rekurrenz-Gleichung

$$a_{n+2} = 2 \cdot a_n - a_{n+1} + 3$$

für die Anfangs-Bedingungen $a_0 = 2$ und $a_1 = 1$.

3.2.3 Lineare inhomogene Rekurrenz-Gleichungen mit veränderlichen Inhomogenitäten

Gelegentlich tauchen in der Praxis Rekurrenz-Gleichungen auf, in denen die Inhomogenität keine Konstante ist, sondern von n abhängt. In solchen Fällen führt die Technik des *diskreten Differenzieren* oft zum Erfolg. Wir stellen die Technik an einem Beispiel vor und betrachten die Rekurrenz-Gleichung

$$a_{n+1} = 2 \cdot a_n + n \quad \text{für alle } n \in \mathbb{N} \tag{3.4}$$

und der Anfangs-Bedingungen $a_0 = 0$. Das Verfahren zur Lösung solcher Rekurrenz-Gleichung besteht aus vier Schritten:

1. Substitutions-Schritt: Im *Substitutions-Schritt* setzen wir in der ursprünglichen Rekurrenz-Gleichung (3.4) für n den Wert $n+1$ ein und erhalten

$$a_{n+2} = 2 \cdot a_{n+1} + n + 1 \quad \text{für alle } n \in \mathbb{N} \tag{3.5}$$

2. Subtraktions-Schritt: Im *Subtraktions-Schritt* ziehen wir von der im Substitutions-Schritt erhaltenen Rekurrenz-Gleichung (3.5) die ursprüngliche gegebene Rekurrenz-Gleichung (3.4) ab. In unserem Fall erhalten wir

$$a_{n+2} - a_{n+1} = 2 \cdot a_{n+1} + n + 1 - (2 \cdot a_n + n) \quad \text{für alle } n \in \mathbb{N}.$$

   Vereinfachung dieser Gleichung liefert

$$a_{n+2} = 3 \cdot a_{n+1} - 2 \cdot a_n + 1 \quad \text{für alle } n \in \mathbb{N}. \tag{3.6}$$

   Die beiden Schritte 1. und 2. bezeichnen wir zusammen als *diskretes Differenzieren* der Rekurrenz-Gleichung.

3. Berechnung zusätzlicher Anfangs-Bedingungen: Die Rekurrenz-Gleichung (3.6) ist eine inhomogene Rekurrenz-Gleichung der Ordnung 2 mit nun aber konstanter Inhomogenität. Wir haben bereits gesehen, wie eine solche Rekurrenz-Gleichung zu lösen ist, wir benötigen aber eine zusätzliche Anfangs-Bedingung für $n = 1$. Diese erhalten wir, indem wir in der ursprünglichen Rekurrenz-Gleichung (3.4) für n den Wert 0 einsetzen:

$$a_1 = 2 \cdot a_0 + 0 = 0.$$

4. Lösen der inhomogenen Rekurrenz-Gleichung mit konstanter Inhomogenität: Das charakteristische Polynom der Rekurrenz-Gleichung (3.6) lautet:

$$\chi(x) = x^2 - 3 \cdot x + 2 = (x-2) \cdot (x-1).$$

   Offenbar gilt $\mathtt{sp}(\chi) = 0$. Um eine spezielle Lösung der Rekurrenz-Gleichung (3.6) zu erhalten, machen wir daher den Ansatz

$$a_n = \varepsilon \cdot n$$

   und erhalten

$$\varepsilon \cdot (n+2) = 3 \cdot \varepsilon \cdot (n+1) - 2 \cdot \varepsilon \cdot n + 1$$

   Diese Gleichung liefert die Lösung

$$\varepsilon = -1.$$

   Damit lautet die allgemeine Lösung der Rekurrenz-Gleichung (3.6):

$$a_n = \alpha_1 \cdot 2^n + \alpha_2 \cdot 1^n - n$$

   Die Koeffizienten α_1 und α_2 finden wir nun durch Einsetzen der Anfangs-Bedingungen:

$$\begin{aligned} 0 &= \alpha_1 + \alpha_2 \\ 0 &= 2 \cdot \alpha_1 + \alpha_2 - 1 \end{aligned}$$

   Aus der ersten Gleichung folgt $\alpha_2 = -\alpha_1$. Damit vereinfacht sich die zweite Gleichung zu

$$0 = 2 \cdot \alpha_1 - \alpha_1 - 1$$

und damit lautet die Lösung $\alpha_1 = 1$ und $\alpha_2 = -1$. Die Lösung der ursprünglichen Rekurrenz-Gleichung (3.4) mit der Anfangs-Bedingung $a_0 = 0$ ist also

$$a_n = 2^n - 1 - n.$$

Das oben gezeigte Verfahren funktioniert, wenn die Inhomogenität der Rekurrenz-Gleichung linear ist, also die Form $\delta \cdot n$. Ist die Inhomogenität quadratisch, so können wir die Gleichung durch diskretes Differenzieren auf eine Rekurrenz-Gleichung reduzieren, deren Inhomogenität linear ist. Diese kann dann aber mit dem eben gezeigten Verfahren gelöst werden. Allgemein gilt: Hat die Inhomogenität der Rekurrenz-Gleichung die Form

$$\delta \cdot n^r \qquad r \in \mathbb{N} \text{ und } r > 0,$$

so kann die Rekurrenz-Gleichung durch r-maliges diskretes Differenzieren auf eine inhomogene Rekurrenz-Gleichung mit konstanter Inhomogenität reduziert werden.

Aufgabe: Lösen Sie die Rekurrenz-Gleichung

$$a_{n+1} = a_n + 2 \cdot n \qquad \text{für alle } n \in \mathbb{N}$$

mit der Anfangs-Bedingung $a_0 = 0$.

Die oben vorgestellte Technik des diskreten Differenzierens führt in leicht variierter Form oft auch dann noch zu einer Lösung, wenn die Inhomogenität nicht die Form eines Polynoms hat. Wir betrachten als Beispiel die Rekurrenz-Gleichung

$$a_{n+1} = a_n + 2^n \quad \text{für alle } n \in \mathbb{N} \tag{3.7}$$

mit der Anfangs-Bedingungen $a_0 = 0$. Setzen wir in (3.7) für n den Wert $n + 1$ ein, erhalten wir

$$a_{n+2} = a_{n+1} + 2^{n+1} \quad \text{für alle } n \in \mathbb{N} \tag{3.8}$$

Würden wir von Gleichung (3.8) die Gleichung (3.7) subtrahieren, so würde der Term 2^n erhalten bleiben. Um diesen Term zu eliminieren müssen wir statt dessen von Gleichung (3.8) 2 mal die Gleichung (3.7) subtrahieren:

$$a_{n+2} - 2 \cdot a_{n+1} = a_{n+1} + 2^{n+1} - 2 \cdot \big(a_n - 2^n\big)$$

Dies vereinfacht sich zu der homogenen Rekurrenz-Gleichung

$$a_{n+2} = 3 \cdot a_{n+1} - 2 \cdot a_n \quad \text{für alle } n \in \mathbb{N} \tag{3.9}$$

Das charakteristische Polynom lautet

$$\chi(x) = x^2 - 3 \cdot x + 2 = (x - 1) \cdot (x - 2).$$

Damit lautet die allgemeine Lösung der homogenen Rekurrenz-Gleichung

$$a_n = \alpha + \beta \cdot 2^n.$$

Da wir hier mit α und β zwei Unbekannte haben, brauchen wir eine zusätzliche Anfangs-Bedingung. Diese erhalten wir, indem wir in der Gleichung (3.7) für n den Wert 0 einsetzen:

$$a_1 = a_0 + 2^0 = 0 + 1 = 1.$$

Damit erhalten wir das Gleichungs-System

$$\begin{array}{rcl} 0 & = & \alpha + \beta \\ 1 & = & \alpha + 2 \cdot \beta \end{array}$$

Dieses Gleichungs-System hat die Lösung $\alpha = -1$ und $\beta = 1$. Damit lautet die Lösung der Rekurrenz-Gleichung (3.7) mit der Anfangs-Bedingung $a_0 = 0$

$$a_n = 2^n - 1.$$

3.2.4 Die Substitutions-Methode

Bei der Analyse von Algorithmen, die dem Paradigma *Teile-und-Herrsche* folgen, treten häufig Rekurrenz-Gleichungen auf, bei denen der Wert von a_n von dem Wert von $a_{n/2}$ oder gelegentlich auch $a_{n/3}$ oder sogar $a_{n/4}$ abhängt. Wir zeigen jetzt ein Verfahren, mit dessen Hilfe sich auch solche Rekurrenz-Gleichungen behandeln lassen. Wir demonstrieren das Verfahren an Hand der Rekurrenz-Gleichung

$$a_n = a_{n/2} + n \quad \text{für alle } n \in \{2^k \mid k \in \mathbb{N} \wedge k \geq 1\} \tag{3.10}$$

mit der Anfangs-Bedingung $a_1 = 0$. Um diese Rekurrenz-Gleichung zu lösen, machen wir den Ansatz

$$b_k = a_{2^k} \quad \text{für alle } k \in \mathbb{N}.$$

Setzen wir dies in die ursprüngliche Rekurrenz-Gleichung (3.10) ein, so erhalten wir

$$b_k = a_{2^k} = a_{2^k/2} + 2^k = a_{2^{k-1}} + 2^k = b_{k-1} + 2^k.$$

Setzen wir in dieser Gleichung für k den Wert $k+1$ ein, so sehen wir, dass die Folge $(b_k)_k$ der Rekurrenz-Gleichung

$$b_{k+1} = b_k + 2^{k+1} \quad \text{für alle } k \in \mathbb{N} \tag{3.11}$$

genügt. Dabei ist die Anfangs-Bedingung $b_0 = a_{2^0} = a_1 = 0$. Das ist eine lineare inhomogene Rekurrenz-Gleichung mit der Inhomogenität 2^{k+1}. Wir setzen in (3.11) für k den Wert $k+1$ ein und erhalten

$$b_{k+2} = b_{k+1} + 2^{k+2} \quad \text{für alle } k \in \mathbb{N}. \tag{3.12}$$

Wir multiplizieren nun die Rekurrenz-Gleichung (3.11) mit 2 und ziehen das Ergebnis von Gleichung (3.12) ab:

$$b_{k+2} - 2 \cdot b_{k+1} = b_{k+1} + 2^{k+2} - 2 \cdot b_k - 2 \cdot 2^{k+1} \quad \text{für alle } k \in \mathbb{N}.$$

Nach Vereinfachung erhalten wir

$$b_{k+2} = 3 \cdot b_{k+1} - 2 \cdot b_k \quad \text{für alle } k \in \mathbb{N}. \tag{3.13}$$

Die Anfangs-Bedingung für $k = 1$ berechnen wir aus (3.11)

$$b_1 = b_0 + 2^1 = 0 + 2 = 2.$$

Damit haben wir das ursprüngliche Problem auf eine homogene lineare Rekurrenz-Gleichung mit konstanten Koeffizienten zurück geführt. Das charakteristische Polynom dieser Rekurrenz-Gleichung ist

$$\chi(x) = x^2 - 3 \cdot x + 2 = (x-2) \cdot (x-1).$$

Damit lautet die allgemeine Lösung der Rekurrenz-Gleichung (3.13)

$$b_k = \alpha_1 \cdot 2^k + \alpha_2 \cdot 1^k \quad \text{für alle } k \in \mathbb{N}.$$

Wir setzen die Anfangs-Bedingungen ein und erhalten so für die Koeffizienten α_1 und α_2 das lineare Gleichungs-System

$$\begin{array}{rcl} 0 & = & \alpha_1 + \alpha_2 \\ 2 & = & 2 \cdot \alpha_1 + \alpha_2 \end{array}$$

Ziehen wir die erste Gleichung von der zweiten ab, so sehen wir $\alpha_1 = 2$. Dann folgt aus der ersten Gleichung $\alpha_2 = -2$. Damit haben wir

$$b_k = 2^{k+1} - 2 \quad \text{für alle } k \in \mathbb{N}.$$

Setzen wir hier $b_k = a_{2^k}$ ein, so finden wir

$$a_{2^k} = 2^{k+1} - 2 \quad \text{für alle } k \in \mathbb{N}.$$

Mit $n = 2^k$ erhalten wir die Lösung der Rekurrenz-Gleichung (3.10) mit der wir gestartet waren:

$$a_n = 2 \cdot n - 2 \quad \text{für alle } n \in \{2^k \mid k \in \mathbb{N}\}.$$

Aufgabe: Lösen Sie die Rekurrenz-Gleichung

$$a_n = a_{n/2} + 1 \quad \text{für alle } n \in \{2^k \mid k \in \mathbb{N} \wedge k \geq 1\}$$

mit der Anfangs-Bedingungen $a_1 = 1$.

3.2.5 Das Teleskop-Verfahren

Bestimmte Rekurrenz-Gleichungen lassen sich auf bereits bekannte Summen zurückführen. Wir demonstrieren das Verfahren an der Rekurrenz-Gleichung

$$a_n = a_{n-1} + n - 1 \quad \text{mit } a_0 = 0.$$

Diese Gleichung tritt bei der Analyse der Komplexität von Quick-Sort auf. Um diese Gleichung zu lösen, setzen wir zunächst für a_{n-1} den Wert $a_{n-2} + (n-1) - 1$ ein, dann ersetzen wir a_{n-2} durch $a_{n-3} + (n-2) - 2$ und fahren so fort, bis wir schließlich a_n auf a_0 zurück geführt haben. Damit erhalten wir insgesamt:

$$\begin{aligned}
a_n &= a_{n-1} + (n-1) \\
&= a_{n-2} + (n-2) + (n-1) \\
&= a_{n-3} + (n-3) + (n-2) + (n-1) \\
&= \vdots \\
&= a_0 + 0 + 1 + 2 + \cdots + (n-2) + (n-1) \\
&= 0 + 0 + 1 + 2 + \cdots + (n-2) + (n-1) \\
&= \sum_{i=0}^{n-1} i \\
&= \tfrac{1}{2} n \cdot (n-1) \\
&= \tfrac{1}{2} n^2 - \tfrac{1}{2}.
\end{aligned}$$

Das eben demonstrierte Verfahren wird in der Literatur als *Teleskop-Verfahren* bezeichnet. In der allgemeinen Form des Teleskop-Verfahrens gehen wir von einer Rekurrenz-Gleichung der Form

$$a_n = a_{n-1} + g(n)$$

aus. Hierbei ist $g : \mathbb{N} \to \mathbb{R}$ eine reelwertige Funktion. Wenden wir das oben demonstrierte Schema an, so erhalten wir die folgende Rechnung:

$$\begin{aligned}
a_n &= a_{n-1} + g(n) \\
&= a_{n-2} + g(n-1) + g(n) \\
&= a_{n-3} + g(n-2) + g(n-1) + g(n) \\
&= \vdots \\
&= a_0 + g(1) + g(2) + \cdots + g(n-2) + g(n-1) + g(n) \\
&= a_0 + \sum_{i=1}^{n} g(i).
\end{aligned}$$

Falls wir in der Lage sind, für die Summe $\sum_{i=1}^{n} g(i)$ einen geschlossenen Ausdruck anzugeben, dann haben wir damit eine Lösung der Rekurrenz-Gleichung $a_n = a_{n-1} + g(n)$ gefunden.

3.2.6 Berechnung von Summen

Der letzte Abschnitt hat gezeigt, dass Rekurrenz-Gleichung in bestimmten Fällen auf Summen zurück geführt werden können. In diesem Abschnitt zeigen wir, dass auch der umgekehrte Weg möglich ist und die Berechnung einer Summe auf die Lösung einer Rekurrenz-Gleichung zurückgeführt werden kann. Wir demonstrieren das Verfahren am Beispiel der Berechnung der geometrischen Reihe. Hier wird die Summe s_n durch die Formel

$$s_n = \sum_{i=0}^{n} q^i \tag{3.14}$$

definiert, wobei wir zur Ersparung von Fallunterscheidungen voraussetzen wollen, dass $q \neq 1$ gilt. Diese Einschränkung ist nicht gravierend denn für $q = 1$ sehen wir sofort, dass $s_n = n+1$ gilt. Der erste Schritt besteht darin, dass wir aus der obigen Definition eine Rekurrenz-Gleichung herleiten. Dies erreichen wir dadurch, dass wir in Gleichung (3.14) für n den Wert $n + 1$ einsetzen. Wir erhalten dann die Gleichung

$$s_{n+1} = \sum_{i=0}^{n+1} q^i \tag{3.15}$$

Wir bilden nun die Differenz von $s_{n+1} - q * s_n$ und erhalten

$$s_{n+1} - s_n \cdot q = 1,$$

was wir zu

$$s_{n+1} = q \cdot s_n + 1$$

umformen. Dies ist eine lineare inhomogene Rekurrenz-Gleichung mit konstanter Inhomogenität. Die Anfangs-Bedingung ist hier offenbar $s_0 = 1$. Das charakteristische Polynom lautet

$$\chi(x) = x - q.$$

Diese Polynom hat die Nullstelle $x = q$. Um die spezielle Lösung der Rekurrenz-Gleichung zu finden, berechnen wir die Spur des charakteristischen Polynoms. Es gilt

$$\texttt{sp}(\chi) = \chi(1) = 1 - q \neq 0,$$

denn wir hatten ja $q \neq 1$ vorausgesetzt. Damit lautet die spezielle Lösung

$$s_n = \frac{c_{-1}}{\texttt{sp}(\chi)} = \frac{1}{1-q}.$$

Folglich lautet die allgemeine Lösung

$$s_n = \alpha \cdot q^n + \frac{1}{1-q}.$$

Um den Koeffizienten α zu bestimmen, setzen wir $n = 0$ und erhalten

$$1 = \alpha + \frac{1}{1-q}.$$

Lösen wir diese Gleichung nach α auf, so ergibt sich

$$\alpha = \frac{(1-q)-1}{1-q} = -\frac{q}{1-q}.$$

Damit lautet die Lösung

$$s_n = \frac{1-q^{n+1}}{1-q}$$

und wir haben insgesamt die folgende Formel hergeleitet:

$$\sum_{i=0}^{n} q^i = \frac{1-q^{n+1}}{1-q}.$$

Aufgabe: Berechnen Sie eine geschlossene Formel für die Summe der Quadratzahlen

$$s_n := \sum_{i=0}^{n} i^2.$$

Stellen Sie dazu eine Rekurrenz-Gleichung für s_n auf und lösen Sie diese.

3.2.7 Weitere Rekurrenz-Gleichungen

Die Lösung allgemeiner Rekurrenz-Gleichungen kann beliebig schwierig sein und es gibt viele Fälle, in denen eine gegebene Rekurrenz-Gleichungen überhaupt keine Lösung hat, die sich durch elementare Funktionen als geschlossene Formel ausdrücken läßt. Wir wollen an Hand einer etwas komplizierteren Rekurrenz-Gleichung, die uns später bei der Behandlung der durchschnittlichen Komplexität des Quick-Sort-Algorithmus wiederbegegnen wird, zeigen, dass im Allgemeinen bei der Lösung einer Rekurrenz-Gleichung Kreativität gefragt ist. Wir gehen dazu von der folgenden Rekurrenz-Gleichung aus:

$$d_{n+1} = n + \frac{2}{n+1} \cdot \sum_{i=0}^{n} d_i. \tag{3.16}$$

Zunächst versuchen wir, die Summe $\sum_{i=0}^{n} d_i$, die auf der rechten Seite dieser Rekurrenz-Gleichung auftritt, zu eliminieren. Wir versuchen, analog zu dem Verfahren des diskreten Differenzierens vorzugehen und substituieren zunächst $n \mapsto n+1$. Wir erhalten

$$d_{n+2} = n + 1 + \frac{2}{n+2} \cdot \sum_{i=0}^{n+1} d_i. \tag{3.17}$$

Wir multiplizieren nun Gleichung (3.17) mit $n+2$ und Gleichung (3.16) mit $n+1$ und haben dann

$$(n+2) \cdot d_{n+2} = (n+2) \cdot (n+1) + 2 \cdot \sum_{i=0}^{n+1} d_i \quad \text{und} \tag{3.18}$$

$$(n+1) \cdot d_{n+1} = (n+1) \cdot n + 2 \cdot \sum_{i=0}^{n} d_i. \tag{3.19}$$

Wir bilden die Differenz der Gleichungen (3.18) und (3.19) und beachten, dass sich die Summationen bis auf den Term $2 \cdot d_{n+1}$ gerade gegenseitig aufheben. Das liefert

$$(n+2) \cdot d_{n+2} - (n+1) \cdot d_{n+1} = (n+2) \cdot (n+1) - (n+1) \cdot n + 2 \cdot d_{n+1}. \tag{3.20}$$

Diese Gleichung vereinfachen wir zu

$$(n+2) \cdot d_{n+2} = (n+3) \cdot d_{n+1} + 2 \cdot (n+1). \tag{3.21}$$

Um diese Gleichung zu homogenisieren teilen wir beide Seiten durch $(n+2) \cdot (n+3)$:

$$\frac{1}{n+3} \cdot d_{n+2} = \frac{1}{n+2} \cdot d_{n+1} + \frac{2 \cdot (n+1)}{(n+2) \cdot (n+3)}. \tag{3.22}$$

Wir definieren $a_n = \dfrac{d_n}{n+1}$ und erhalten dann aus der letzten Gleichung

$$a_{n+2} = a_{n+1} + \frac{2 \cdot (n+1)}{(n+2) \cdot (n+3)}$$

Die Substitution $n \mapsto n-2$ vereinfacht diese Gleichung zu

$$a_n = a_{n-1} + \frac{2 \cdot (n+1)}{(n+2) \cdot (n+3)}. \tag{3.23}$$

Diese Gleichung können wir mit dem Teleskop-Verfahren lösen. Um die dabei auftretenden Summen übersichtlicher schreiben zu können, bilden wir die Partialbruch-Zerlegung von

$$\frac{2\cdot(n+1)}{(n+2)\cdot(n+3)}.$$

Dazu machen wir den Ansatz

$$\frac{2\cdot(n+1)}{(n+2)\cdot(n+3)} = \frac{\alpha}{n+2} + \frac{\beta}{n+3}.$$

Wir multiplizieren diese Gleichung mit dem Hauptnenner und erhalten

$$2\cdot n+2 = \alpha\cdot(n+3) + \beta\cdot(n+2),$$

was sich zu

$$2\cdot n+2 = (\alpha+\beta)\cdot n + 3\cdot\alpha + 2\cdot\beta$$

vereinfacht. Ein Koeffizientenvergleich liefert dann das lineare Gleichungs-System:

$$\begin{array}{rcl} 2 & = & \alpha+\beta \\ 2 & = & 3\cdot\alpha+2\cdot\beta \end{array}$$

Ziehen wir die erste Gleichung zweimal von der zweiten Gleichung ab, so erhalten wir $\alpha = -2$ und Einsetzen in die erste Gleichung liefert $\beta = 4$. Damit können wir die Gleichung (3.23) als

$$a_n = a_{n-1} - \frac{2}{n} + \frac{4}{n+1} \tag{3.24}$$

schreiben und mit dem Teleskop-Verfahren lösen. Wegen $a_0 = \frac{d_0}{1} = 0$ finden wir

$$a_n = 4\cdot\sum_{i=1}^{n}\frac{1}{i+1} - 2\cdot\sum_{i=1}^{n}\frac{1}{i}. \tag{3.25}$$

Wir vereinfachen diese Summe:

$$\begin{array}{rcl} a_n & = & 4\cdot\sum\limits_{i=1}^{n}\frac{1}{i+1} - 2\cdot\sum\limits_{i=1}^{n}\frac{1}{i} \\ & = & 4\cdot\sum\limits_{i=2}^{n+1}\frac{1}{i} - 2\cdot\sum\limits_{i=1}^{n}\frac{1}{i} \\ & = & 4\cdot\frac{1}{n+1} - 4\cdot\frac{1}{1} + 4\cdot\sum\limits_{i=1}^{n}\frac{1}{i} - 2\cdot\sum\limits_{i=1}^{n}\frac{1}{i} \\ & = & 4\cdot\frac{1}{n+1} - 4\cdot\frac{1}{1} + 2\cdot\sum\limits_{i=1}^{n}\frac{1}{i} \\ & = & -\frac{4\cdot n}{n+1} + 2\cdot\sum\limits_{i=1}^{n}\frac{1}{i} \end{array}$$

Um unsere Rechnung abzuschließen, berechnen wir eine Näherung für die Summe

$$H_n = \sum_{i=1}^{n}\frac{1}{i}.$$

Der Wert H_n wird in der Mathematik als die n-te *harmonische Zahl* bezeichnet. Dieser Wert hängt mit dem Wert $\ln(n)$ zusammen: Leonhard Euler hat gezeigt, dass für große n die Approximation

$$\sum_{i=1}^{n}\frac{1}{i} \approx \ln(n)$$

benutzt werden kann.

Wir verallgemeinern die Idee, die wir bei der Lösung des obigen Beispiels benutzt haben. Es seien $f : \mathbb{N} \to \mathbb{R}$, $g : \mathbb{N} \to \mathbb{R}$ und $h : \mathbb{N} \to \mathbb{R}$ reelwertige Folgen und es sei die Rekurrenz-Gleichung

$$f(n) \cdot a_n = g(n) \cdot a_{n-1} + h(n)$$

zu lösen. Die Idee ist, beide Seiten mit einem geeigneten Faktor, der im Allgemeinen von n abhängt, zu multiplizieren. Bezeichnen wir diesen Faktor mit $p(n)$, so erhalten wir die Rekurrenz-Gleichung

$$p(n) \cdot f(n) \cdot a_n = p(n) \cdot g(n) \cdot a_{n-1} + p(n) \cdot h(n).$$

Das Ziel ist dabei, den Faktor $p(n)$ so zu wählen, dass der Koeffizient von a_n die selbe Form hat wie der Koeffizient von a_{n-1}, es soll also

$$p(n) \cdot g(n) = p(n-1) \cdot f(n-1) \tag{3.26}$$

gelten, denn dann können wir die ursprüngliche Rekurrenz-Gleichung in der Form

$$p(n) \cdot f(n) \cdot a_n = p(n-1) \cdot f(n-1) \cdot a_{n-1} + p(n) \cdot h(n).$$

schreiben und anschließend durch die Substitution $b_n := p(n) \cdot f(n) \cdot a_n$ auf die Rekurrenz-Gleichung

$$b_n = b_{n-1} + p(n) \cdot h(n).$$

Diese Gleichung läßt sich mit dem Teleskop-Verfahren auf eine Summe zurückführen und die Lösung der ursprünglichen Gleichung kann schließlich über die Formel

$$a_n = \frac{1}{p(n) \cdot f(n)} \cdot b_n$$

aus b_n berechnet werden. Es bleibt also zu klären, wie wir den Faktor $p(n)$ so wählen können, dass Gleichung (3.26) erfüllt ist. Dazu schreiben wir diese Gleichung als Rekurrenz-Gleichung für $p(n)$ um und erhalten

$$p(n) = \frac{f(n-1)}{g(n)} \cdot p(n-1)$$

Diese Gleichung können wir mit einer Variante des Teleskop-Verfahrens lösen:

$$\begin{array}{rcl}
p(n) & = & \frac{f(n-1)}{g(n)} \cdot p(n-1) \\
 & = & \frac{f(n-1)}{g(n)} \cdot \frac{f(n-2)}{g(n-1)} \cdot p(n-2) \\
 & = & \frac{f(n-1)}{g(n)} \cdot \frac{f(n-2)}{g(n-1)} \cdot \frac{f(n-3)}{g(n-2)} \cdot p(n-3) \\
 & = & \frac{f(n-1)}{g(n)} \cdot \frac{f(n-2)}{g(n-1)} \cdot \frac{f(n-3)}{g(n-2)} \cdot p(n-3) \\
 & \vdots & \\
 & = & \frac{f(n-1)}{g(n)} \cdot \frac{f(n-2)}{g(n-1)} \cdot \frac{f(n-3)}{g(n-2)} \cdot \dots \cdot \frac{f(2)}{g(3)} \cdot \frac{f(1)}{g(2)} \cdot p(1)
\end{array}$$

Wir setzen willkürlich $p(1) = 1$ und haben dann für $p(n)$ die Lösung

$$p(n) = \prod_{i=1}^{n-1} \frac{f(i)}{g(i+1)}$$

gefunden. Bei der Rekurrenz-Gleichung

$$n \cdot d_n = (n+1) \cdot d_{n-1} + 2 \cdot (n-1),$$

die aus der Rekurrenz-Gleichung (3.21) durch die Substitution $n \mapsto n-2$ hervorgeht, gilt $f(n) = n$ und $g(n) = n + 1$. Damit haben wir dann

$$\begin{aligned}
p(n) &= \prod_{i=1}^{n-1} \tfrac{f(i)}{g(i+1)} \\
&= \prod_{i=1}^{n-1} \tfrac{i}{i+2} \\
&= \tfrac{1}{3} \cdot \tfrac{2}{4} \cdot \tfrac{3}{5} \cdot \cdots \cdot \tfrac{n-3}{n-1} \cdot \tfrac{n-2}{n} \cdot \tfrac{n-1}{n+1} \\
&= 2 \cdot \tfrac{1}{n} \cdot \tfrac{1}{n+1}.
\end{aligned}$$

Die Konstante 2 ist hier unwichtig und wir sehen, dass der Faktor $\frac{1}{n\cdot(n+1)}$ benutzt werden kann, um die ursprüngliche Rekurrenz-Gleichung zu homogenisieren.

Aufgabe 1: Lösen Sie die Rekurrenz-Gleichung

$$a_n = 2 \cdot a_{n-1} + 1 \quad \text{mit } a_0 = 0$$

mit Hilfe einer geeigneten Homogenisierung. Gehen Sie dabei analog zu dem im letzten Abschnitt beschriebenen Verfahren vor.