§ 1. Die Fibonacci-Zahlen

1.1. Definition. Die Folge der *Fibonacci-Zahlen* $(f_n)_{n\geqslant 0}$ wird rekursiv definiert durch

$$f_0 = 0, \quad f_1 = 1 \quad \text{und} \quad f_{n+2} = f_{n+1} + f_n \quad \text{für alle } n \geqslant 0.$$

Von der zweiten Stelle an ist also jedes Glied der Folge gleich der Summe der beiden vohergehenden. Die ersten Fibonacci-Zahlen sind

n	0	1	2	3	4	5	6	7	8	9	10	11	12
f_n	0	1	1	2	3	5	8	13	21	34	55	89	144

1.2. Als erstes leiten wir eine explizite Formel für die Fibonacci-Zahlen her. Dazu untersuchen wir allgemein Folgen (x_n), die der Rekursionsformel

$$x_{n+2} = x_{n+1} + x_n \quad \text{für alle } n \geqslant 0 \tag{1}$$

genügen (ohne eine Anfangsbedingung festzulegen) und machen dafür den Ansatz $x_n := \lambda^n$ mit einer noch zu bestimmenden reellen (oder komplexen) Zahl λ. Die Rekursionsbedingung lautet dann

$$\lambda^{n+2} = \lambda^{n+1} + \lambda^n$$

und ist offenbar erfüllt, falls

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

Diese Gleichung hat die Lösungen

$$\lambda = \frac{1+\sqrt{5}}{2} \approx 1.61803\ldots \quad \text{und} \quad \lambda' = \frac{1-\sqrt{5}}{2} = 1 - \lambda = -\frac{1}{\lambda}.$$

Die Zahl λ ist der berühmte *goldene Schnitt*.
Es folgt, dass für beliebige Konstanten c_1, c_2 auch die Folge

$$x_n := c_1\,\lambda^n + c_2\,(\lambda')^n, \quad n \geq 0,$$

der Rekursionsformel (1) genügt. Man kann jetzt die Konstanten c_1, c_2 so anpassen, dass die Anfangsbedingungen für die Fibonacci-Zahlen erfüllt sind. Dies führt auf das Gleichungs-System

$$\begin{aligned} 0 &= c_1 + c_2, \\ 1 &= c_1\lambda + c_2(1-\lambda) \end{aligned}$$

mit der Lösung

$$c_1 = \frac{1}{2\lambda - 1} = \frac{1}{\sqrt{5}} = -c_2.$$

Damit haben wir folgenden Satz bewiesen:

1.3. Satz. *Mit dem goldenen Schnitt $\lambda = \frac{1}{2}(1 + \sqrt{5})$ gilt für die n-te Fibonacci-Zahl die Formel*

$$f_n = \frac{1}{\sqrt{5}}\Big(\lambda^n - \frac{(-1)^n}{\lambda^n}\Big).$$

Da für $n \geq 0$ stets $|\lambda^{-n}/\sqrt{5}| < \frac{1}{2}$ ist, folgt daraus, dass

$$f_n = \operatorname{round}\Big(\frac{\lambda^n}{\sqrt{5}}\Big),$$

wobei $\operatorname{round}(x)$ die der reellen Zahl x nächstgelegene ganze Zahl bedeutet.
Aus der Formel erkennt man das exponentielle Wachstum der Fibonacci-Zahlen. Da für den Logarithmus zur Basis 10 des goldenen Schnitts gilt

$$\log_{10} \lambda \approx 0.20898\ldots,$$

hat die n-te Fibonacci-Zahl etwa $0.209 \cdot n \approx n/4.78$ Dezimalstellen. Einige spezielle Werte sind

$$\begin{aligned} f_{10} &= 55, \\ f_{20} &= 6765, \\ f_{50} &= 1\ 25862\ 69025, \\ f_{100} &= 3\ 54224\ 84817\ 92619\ 15075, \\ f_{200} &= 28\ 05711\ 72992\ 51014\ 00376\ 11932\ 41303\ 86771\ 89525. \end{aligned}$$

1.4. Die Formel von Satz 3 ist zwar insofern interessant, als sie die ganzzahlige Folge der Fibonacci-Zahlen mit den Potenzen einer irrationalen Zahl, dem goldenen Schnitt λ, in Verbindung bringt, ist aber für zahlentheoretische Untersuchungen weniger zu gebrauchen. So fällt bei den oben angegebenen Werten auf, dass f_{10} ein Teiler von f_{20} ist, denn 6765 ist sowohl durch 5 wie auch durch 11 ohne Rest teilbar. Ebenso ist f_{100} ein Teiler von f_{200}, was aber nicht mehr mit bloßem Auge zu sehen ist. Um Aussagen dieser Art beweisen zu können (z.B. für alle k ist f_k ein Teiler von f_{2k}) ist eine andere Darstellung der Fibonacci-Zahlen nützlich, die wir jetzt ableiten.
Das Gleichungssystem

$$\begin{aligned} f_{n+1} &= f_n + f_{n-1} \\ f_n &= f_n \end{aligned}$$

lässt sich in Matrizen-Schreibweise so darstellen:

$$\begin{pmatrix} f_{n+1} \\ f_n \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} f_n \\ f_{n-1} \end{pmatrix}.$$

Dies gilt übrigens auch für $n = 0$, wenn man $f_{-1} := 1$ definiert. Setzt man

$$F_n := \begin{pmatrix} f_n \\ f_{n-1} \end{pmatrix} \quad \text{und} \quad A := \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix},$$

so schreibt sich die Gleichung einfach

$$F_{n+1} = AF_n \quad \text{für alle } n \geq 0.$$

In die letzte Gleichung kann man $F_n = AF_{n-1}$ substituieren, u.s.w. Durch vollständige Induktion erhält man

$$F_{n+1} = A^n F_1 \quad \text{und} \quad F_n = A^n F_0 \tag{2}$$

Die beiden Spaltenvektoren F_{n+1} und F_n kann man zu einer 2×2-Matrix

$$(F_{n+1} \;\; F_n) = \begin{pmatrix} f_{n+1} & f_n \\ f_n & f_{n-1} \end{pmatrix}$$

zuammenfassen. Damit schreibt sich (2) als

$$(F_{n+1} \;\; F_n) = A^n \, (F_1 \;\; F_0) = A^n \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = A^n.$$

Wir formulieren das Ergebnis als Satz.

1.5. Satz. *Für alle $n \geq 0$ gilt die Matrizengleichung*

$$\begin{pmatrix} f_{n+1} & f_n \\ f_n & f_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n.$$

Als ein Beispiel für die Anwendung dieses Satzes beweisen wir:

Folgerung. Für jede natürliche Zahl $k \geqslant 1$ ist f_k ein Teiler von f_{2k}; genauer gilt

$$f_{2k} = f_k(f_{k+1} + f_{k-1}) = f_k(f_k + 2f_{k-1}).$$

Beweis. Die Formel des Satzes liefert

$$\begin{pmatrix} f_{2k+1} & f_{2k} \\ f_{2k} & f_{2k-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^k \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^k = \begin{pmatrix} f_{k+1} & f_k \\ f_k & f_{k-1} \end{pmatrix} \begin{pmatrix} f_{k+1} & f_k \\ f_k & f_{k-1} \end{pmatrix}.$$

Durch Matrizen-Multiplikation erhält man daraus

$$f_{2k} = f_k f_{k+1} + f_{k-1} f_k,$$

woraus die Behauptung folgt.

Etwas allgemeiner kann man zeigen (Übung):

$$\textit{Für alle } k \geqslant 1 \textit{ und alle } m \geqslant 1 \textit{ ist } f_k \textit{ ein Teiler von } f_{mk}. \tag{3}$$

Daraus folgt, dass f_n, $n > 4$, höchstens dann eine Primzahl ist, wenn der Index n prim ist. Die Umkehrung gilt aber nicht. Tatsächlich sind z.B.

$$f_5 = 5,\ f_7 = 13,\ f_{11} = 89,\ f_{13} = 233,\ f_{17} = 1597,\ f_{23} = 28657$$

Primzahlen, aber

$$f_{19} = 4181 = 37 \cdot 113$$

ist nicht prim. $f_4 = 3$ ist eine Primzahl, obwohl der Index 4 nicht prim ist. Da $f_2 = 1$, ist dies kein Widerspruch zu der obigen Teilbarkeits-Aussage (3).

1.6. Ein schneller Potenzierungs-Algorithmus. Die naheliegende Methode zur Berechnung einer Potenz A^N ist, durch Multiplikation mit A der Reihe nach $A^2, A^3, \ldots, A^N$ auszurechnen. Dabei braucht man $N - 1$ Multiplikationen. Dass diese Methode nicht immer die günstigste ist, sieht man z.B. im Fall $N = 64$, wo man durch sukzessive Quadrierungen

$$A \to A^2 \to A^4 \to A^8 \to A^{16} \to A^{32} \to A^{64}$$

mit 6 statt mit 63 Multiplikationen auskommt. Diese Idee lässt sich auch auf Exponenten, die keine reine Zweier-Potenz sind, verallgemeinern. Dies beruht auf dem folgenden Lemma.

Lemma. *Sei N eine natürliche Zahl mit $2^d \leqslant N < 2^{d+1}$. Dann gibt es eine Kette*

$$1 = n_0 < n_1 < n_2 < \ldots < n_{d-1} < n_d = N$$

natürlicher Zahlen, wobei für alle $k = 0, 1, \ldots, d-1$ gilt

$$n_{k+1} = \begin{cases} 2n_k & \text{oder} \\ 2n_k + 1 & \end{cases}$$

Beweis. Wir benutzen die Binär-Darstellung von N,

$$N = \sum_{k=0}^{d} b_k \cdot 2^k, \quad b_d = 1,\ b_k \in \{0, 1\} \text{ für } k = 0, \ldots, d-1.$$

Wir kürzen dies ab als

$$N = (b_d b_{d-1} \ldots b_1 b_0)_2.$$

Die Zahlen n_k werden nun definiert als

$$n_k := (b_d \dots b_{d-k})_2$$

Offensichtlich gilt $n_0 = 1$ und $n_d = N$ sowie

$$n_{k+1} = (b_d \dots b_{d-k} b_{d-k-1})_2 = 2n_k + b_{d-k-1}, \quad \text{q.e.d.}$$

Corollar. *Sei N eine natürliche Zahl mit $2^d \leqslant N < 2^{d+1}$. Dann lässt sich A^N mit d Quadrierungen und höchstens d Multiplikationen berechnen.*

Beweis. Sei $1 = n_0 < n_1 < n_2 < \dots < n_{d-1} < n_d = N$ eine Kette natürlicher Zahlen wie im Lemma. Wir berechnen der Reihe nach

$$A = A^{n_0} \to A^{n_1} \to A^{n_2} \to \dots \to A^{n_d} = A^N.$$

Für jedes $k > 0$ gilt

$$A^{n_k} = (A^{n_{k-1}})^2 \quad \text{oder} \quad A^{n_k} = (A^{n_{k-1}})^2 A.$$

Daraus folgt die Behauptung.

Zusammen mit Satz 1.5 lässt sich daraus ein schneller Algorithmus zur Berechnung der Fibonacci-Zahlen konstruieren.

In den Anwendungen der Zahlentheorie treten öfter Potenzen mit großem Exponenten auf, wo man den schnellen Potenzierungs-Algorithmus gut gebrauchen kann.