

– Praktikumsaufgabe 3 –

Thema: *Elementare Konstrukte in Rust*

Zielstellung: Praktische Arbeit mit Datentypen, Schleifen und Funktionen

1. Entwickeln Sie ein Rust-Programm, das die Wertebereiche aller Ganzzahl-Typen ermittelt.
2. Die Folge f der Fibonacci-Zahlen ist definiert durch

$$f(n) = \begin{cases} 1 & \text{für } n = 0, \\ 1 & \text{für } n = 1, \\ f(n-1) + f(n-2) & \text{ansonsten.} \end{cases}$$

Schreiben Sie eine Rust-Funktion, die n als Parameter übernimmt und *rekursiv* (gemäß obiger Definition) $f(n)$ ermittelt. Im Hauptprogramm implementieren Sie eine Schleife, die die Funktion mit aufsteigendem n aufruft. Welche Tiefe erreichen Sie und wie groß ist die Laufzeit (Shellkommando `time`)? **uses 18% cpu, 0,03s**

3. Das Sieb des Erathostenes ist ein (ur)alter Algorithmus zur Bestimmung einer großen Menge Primzahlen. Nutzen Sie ein Array und implementieren Sie das Sieb, um alle Primzahlen zwischen 2 und 100.000 zu ermitteln. **Hinweise:**
 - Sie müssen alle Vielfachen der Primzahlen zwischen 2 und $\sqrt{1.000.000}$ aussieben. Die obere Grenze können Sie folgendermaßen darstellen (`MAXIMUM` ist eine `usize`-Konstante):
`(MAXIMUM as f64).sqrt() as usize`
- 4.* Die so genannte *Ackermannfunktion* ist folgendermaßen definiert¹:

$$\begin{aligned} a(0, m) &= m + 1 \\ a(n + 1, 0) &= a(n, 1) \\ a(n + 1, m + 1) &= a(n, a(n + 1, m)) \end{aligned}$$

Im zugehörigen Wikipedia-Eintrag findet sich, bezogen auf $m = 3$, die folgende Aussage:

Mit Java 1.4.2 und den Standardspeichereinstellungen erreicht man heutzutage $n = 13$.

Überprüfen Sie, ob Sie mit Rust und einem Standard-Laborrechner ebenfalls $a(3, 13)$ ermitteln können. Schaffen Sie auch $a(3, 14)$? Wie lange dauert das?

¹Die angegebene Definition ist nach Rózsa Péter vereinfacht.