

Computermathematik – Übung Sage3

- **Abgabeschluss:** Di 04.12.2018 um 14:00
- **Präsentation:** Mi 05.12.2018
- **Abgabeformat:** Sage3.zip / Aufgabe11.sws, Aufgabe12.sws, Aufgabe13.sws.

Aufgabe 11 – Primzahlen 2+1+1 Punkte)

- (a) Bestimme die Anzahl aller 6-stelligen Primzahlen mit den folgenden Verfahren und vergleiche die Laufzeiten (mit `cputime` oder `timeit`):
- Unter Verwendung einer `for`-Schleife und `Integer.is_prime()`.
 - Unter Verwendung einer `while`-Schleife und `Integer.is_prime()`.
 - Unter Verwendung einer `while`-Schleife und `Integer.next_prime()`.
 - Unter Verwendung von `prime_range`.
- (b) Modifiziere anschließend den Code jeweils so, daß eine Liste der Primzahlen erstellt wird.
- (c) Erstelle eine Liste aller 6-stelligen Primzahlzwillinge (d.h., Paare von Primzahlen (p, q) , sodaß $q = p + 2$).

Aufgabe 12 – Goldbachsche Vermutung (2+1 Punkte)

Zeige, daß jede gerade Zahl zwischen 4 und 10000 als Summe zweier Primzahlen geschrieben werden kann (Goldbachsche Vermutung).

- (a) Schreibe eine Funktion `goldbach(n)`, die zu gegebenem n eine Liste der Zerlegungen berechnet.
- (b) Welche Zahl hat die meisten Zerlegungen?
- (c) Erstelle eine Graphik mit `bar_chart`.

Hinweise: `Integer.is_prime()`, `Integer.next_prime()`.

Aufgabe 13 – Approximation des Logarithmus (2+2 Punkte)

Der natürliche Logarithmus kann durch die folgende Reihe approximiert werden:

$$\ln x \approx 2 \sum_{k=0}^n \frac{(x-1)^{2k+1}}{(2k+1)(x+1)^{2k+1}}$$

- (a) Implementiere in Sage eine Funktion `s1(x,n)`, die die obige Summe im Punkt x auswertet.
- (b) Implementiere in Sage eine Funktion `s2(x,δ)`, die die obige Summe solange aufsummiert, als der Absolutbetrag des neuen Summanden größer ist als δ .

Vergleiche die Werte der Funktionen mit $\ln(x)$ für die Werte $x = 10^k$, $k \in \{-5, -4, \dots, 4, 5\}$ und verschiedene Werte von n und δ .