Mersenne-Primzahlen

lucasl.py

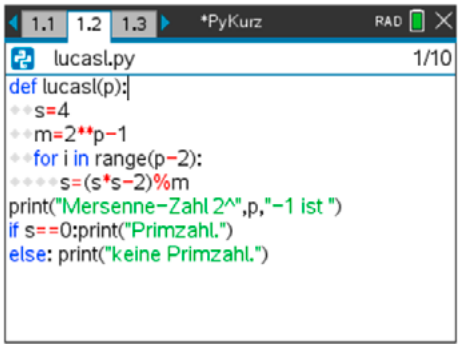
Eine Mersenne-Zahl ist eine natürliche Zahl der Form $M_n = 2^n - 1$. Eine Mersenne-Primzahl ist demnach eine Mersenne-Zahl, die außerdem auch eine Primzahl ist.

Wenn n keine Primzahl ist, dann ist auch M_n keine. Das folgt aus der Tatsache, dass $2^{a\,b} - 1 = (2^a - 1)(1 + 2^a + 2^{2a} + 2^{3a} + \ldots + 2^{(b\text{-}1)a})$. Wenn n prim ist, muss M_n nicht auch prim sein. So ist zum Beispiel $M_{11} = 2^{11} - 1 = 2047 = 23 \cdot 89$.

Es gibt nun einen einfachen Test (Lucas-Lehmer-Test), der überprüft, ob für eine Primzahl $p > 2$, die Mersenne-Zahl $M_p = 2^p - 1$ eine Primzahl ist. Dafür bilden wir die Folge

$$s_i = s_{i-1}^2 - 2 \bmod M_p \quad \text{mit } s_0 = 4.$$

Wenn nun $s_{p-2} = 0$, dann ist M_p eine Primzahl. Das lässt sich in Python einfach implementieren.



- Wir definieren die Prozedur lucasl(p), die das Argument $p > 2$ und prim erwartet.
- Wir setzen $s = 4$ und berechnen $m = M_p = 2^p - 1$.
- In einer Schleife wird $p - 2$ mal der nachfolgende s-Wert mit s = (s*s–2)%m berechnet.
- Dann wird die Mersenne-Zahl ausgegeben gemeinsam mit der Information, ob sie eine Primzahl ist ($s = 0$) oder nicht.

Mersenne-Zahlen werden sehr rasch sehr groß. Daher führt dieser einfache Test zu einer ganzen Folge von sehr großen Primzahlen. Seit 2005 ist die größte bekannte Primzahl immer eine Mersenne-Zahl gewesen.

https://de.wikipedia.org/wiki/Great_Internet_Mersenne_Prime_Search
http://www.mathe.tu-freiberg.de/~hebisch/cafe/mersenneprim.html