kry.tns Bibliotheks-Datei für Kryptografie Haftendorn Okt 2011

Vorhandene Befehle: $\mathrm{mod}(a,\mathbf{m})$ ist a modulo m, also $\mathrm{mod}(54,7) \to 5$ (Eingebauter Befehl)

Größter gemeinamer Teiler $\gcd(54,30) \to 6$ Der erweiterte euklidische Algorithmus ist **programmiert**: $\mathbf{ggte}(54,30) \to [6 \quad -1 \quad 2]$ Seite 4 ausführlich .

Powermod ist der wichtigste Befehl. $\mathbf{pmod}(a,k,\mathbf{m})$ ist a^k *modulo* $\mathbf{m}$

Mit $\mathbf{m}:=7 \to 7$ also $\mathbf{pmod}(2,5,7) \to 4$ .Das ist dasselbe wie $\mathrm{mod}(2^5,7) \to 4$, nur pmod ist für sehr große Zahlen möglich.

$\mathrm{mod}(12345^{6789},7) \to \mathrm{mod}(\infty,7)$ ⚠ geht nicht, aber $\mathbf{pmod}(12345,6789,7) \to 1$ klappt.

$\mathbf{ordo}(a,\mathbf{m})$ berechnet die Ordnung von a in Z*(m), $\mathbf{ordo}(5,13) \to 4$ sagt vorher: $\mathrm{mod}(5^4,13) \to 1$

$\mathbf{maltafel}(6) \to \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 0 & 2 & 4 \\ 3 & 0 & 3 & 0 & 3 \\ 4 & 2 & 0 & 4 & 2 \\ 5 & 4 & 3 & 2 & 1 \end{bmatrix}$ $\mathbf{maltafel}(7) \to \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 6 & 1 & 3 & 5 \\ 3 & 6 & 2 & 5 & 1 & 4 \\ 4 & 1 & 5 & 2 & 6 & 3 \\ 5 & 3 & 1 & 6 & 4 & 2 \\ 6 & 5 & 4 & 3 & 2 & 1 \end{bmatrix}$ zeigt Multiplikationstafeln

Die Nullen innerhalb der Tafeln stören, darum nimmt man nur teilerfremde Zahlen:

$\mathbf{zstern}(\mathbf{m}) \to \{1,2,3,4,5,6\}$ zeigt die Menge der zu m teilerfremden Zahlen. $\mathbf{zstern}(10) \to \{1,3,7,9\}$

$\mathbf{malstern}(\mathbf{m}) \to \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 6 & 1 & 3 & 5 \\ 3 & 6 & 2 & 5 & 1 & 4 \\ 4 & 1 & 5 & 2 & 6 & 3 \\ 5 & 3 & 1 & 6 & 4 & 2 \\ 6 & 5 & 4 & 3 & 2 & 1 \end{bmatrix}$ gibt die Maltafel von Z*(m) $\mathbf{malstern}(10) \to \begin{bmatrix} 1 & 3 & 7 & 9 \\ 3 & 9 & 1 & 7 \\ 7 & 1 & 9 & 3 \\ 9 & 7 & 3 & 1 \end{bmatrix}$

$\mathbf{eulerphi}(\mathbf{m}) \to 6$ gibt die Anzahl der Elemente von z*(m) an, also die Zahl der zu m teilerfremden $\mathbf{eulerphi}(10) \to 4$

Weitere Befehle:

eingebaut ist: $\mathrm{isPrime}(71) \to \mathrm{true}$ $\mathrm{isPrime}(120000000000031) \to \mathrm{true}$

$\mathrm{factor}(91) \to 7 \cdot 13$ $\mathrm{factor}(71) \to 71$ $\mathrm{factor}(120000000000031 \cdot 1-2) \to 1433849 \cdot 83690821$

In dieser Datei **programmierte Befehle**

$\mathbf{nextprime}(120000000000000)$ die Angabe der nächst größeren Primzahl

In der Bibliothek ist noch $\mathbf{istprim\backslash istprim}(1729,1) \to \mathrm{true}$ $\mathbf{istprim\backslash istprim}(1729,10) \to \mathrm{false}$

der Miller-Rabin-Test programmiert, er entlarvt auch Pseudoprimzahlen.

Siehe Erkläsungsseite zu Pseudoprimzahlen.

teiler(m) ▸ {1,7} gibt alle Teiler von m an **teiler(24)** ▸ {1,2,3,4,6,8,12,24}

Das sind –außer der 1– gerade die Zahlen, die in **zstern(24)** ▸ {1,5,7,11,13,17,19,23} fehlen.

Für das Verstehen von Kryptografie sind vor allem die Potenzen in Z*(m) wichtig

potstern(18) ▸

1	1	5	7	11	13	17
2	1	7	13	13	7	1
3	1	17	1	17	1	17
4	1	13	7	7	13	1
5	1	11	13	5	7	17
6	1	1	1	1	1	1

Die erste Spalte gibt den Exponenten k an,

mod(11^3,18) ▸ 17

Die erste Zeile (ab Platz 2) ist die Basis a , innen steht dann a^k *modulo* **m** .Die Ordnung von a ist die Zeilennummer k der als erstes von oben nach unten auftauchenden 1.

Der Eulersche Satz besagt: in den Potenztafeln von Z*(m) steht in der letzen Zeile überall 1.

*Stelle die **kry.tns** in das Verzeichnis MyLib auf dem Computer oder dem Handheld. Mache "Bibliotheken aktualisieren" (Handheld: Menu 1–7–1, PC Extras)*

*Arbeiten damit: Handeld: Taste Buch 6 (Bib.) PC: Buch, unten Bibliotheken. Klappe "Bibliotheken" auf, **Beide:** wähle kry und greife die benötigten Befehle z.B.* $kryp\backslash pmod()$

Erweiterter Euklidischer Algorithmus und Bestimmung des Inversen modulo m

ggte(a,b) ergibt die Zahlen [g,a,b] der Vielfach–Summen–Darstellung $g=s\cdot a+t\cdot b$ g ist dabei der größte gemeinsame Teiler. **ggte**(16,21) ▸ [1 4 -3] ist also zu deuten als

1=4·16+-3·21 ▸ true

Dieses nützt für die Suche nach dem multiplikativ Inversen modulo b (oder a). Obige Gleichung modulo 21 betrachtet ergibt $1=4\cdot 16$ modulo 21, also ist 4 das Inverse von 16 in Z*(21) Probe mod(4·16,21) ▸ 1

Man kann die Gleichung auch modulo 16 nutzen:$1=-3\cdot 21$ modulo 16. Zu negativen Zahlen addiert einmal die Modul–Zahl m, also $1=13\cdot 21$ modulo 16, Probe mod(13·21,16) ▸ 1

Das Verfahren, das in ggte programmiert ist, heißt: erweiterter eulidischer Algorithmus. Er arbeitet auch für die riesigen Zahlen der Kryptografie effektiv.

Die Zahlen s und t heißen auch "Bézout Koeffizienten", ihre Existenz sichert das "**Lemma von Bézout**"

Siehe Wikipedia. Man kann es aber einfach (schulisch sinnvoll) begründen durch Rückwärtsarbeiten mit dem Euklidischen Algorithmus.