Ägyptischens Rechnen und Stammbrüche

Ägyptische Multiplikation

Ägyptisches Rechnen   Haftendorn 2009 und 2011

			413
413	12	1	1
~~206~~	~~24~~	0	~~2~~
103	48	1	4
51	96	1	8
25	192	1	16
~~12~~	~~384~~	0	~~32~~
~~6~~	~~768~~	0	~~64~~
3	1536	1	128
1	3072	1	256
	Σ 4956		413

Bei der ägyptischen Multiplikation wird der 1. Faktor fortgesetzt ohne Rest halbiert, der zweite Faktor wird verdoppelt. Die Zeilen der geraden Zahlen aus der 1. Spalte werden gestrichen, die verbleibenden Zahlen der zweiten Spalte werden addiert. Diese Summe ist das gesuchte Produkt. Diese Handlingsweise eintspricht der Erzeugung des 1. Faktors aus Binärzahl mit der DoubleDaddel-Methode.In der 2. Spalte stehen die 12-fachen der bei 413 verwendente 2-Potenzen 413▸Base2 • 0b110011101   die rechte Ziffer steht also oben.

$256+128+16+8+4+1$ • 413 $12\cdot(256+128+16+8+4+1)$ • 4956 $413\cdot 12$ • 4956

Die Halbierung ohne Rest wird von **halb**$(x):=\text{floor}\left(\frac{x}{2}\right)$ • *Fertig* erledigt. **halb**(413) • 206

1.1

Programmierte Funktionen

nim$(x,y):=\begin{cases} y, \text{mod}(x,2)=1 \\ 0, \text{mod}(x,2)=0 \end{cases}$ • *Fertig* **nim**$(45,30)$ • 30 **nim**$(20,30)$ • 0

Diese Funktion betrachtet in jeder Zeile x aus der 1. und y aus der 2. Spalte und sorgt dafür dass die linken Einträge rechts neben ungerade Zahlen für die spätere Summation genommen werden. Es bietet sich an, die Rechnungenim Tabellen-Fensterdurchzuführen.

Im Tabbellenfesnter vom TI Nspire werden die Namen der Spalten als Varablen betrachtet.

Darum kann man sie Summen von Spalten (z.B.) nicht unter den betreffenden Zahlen sondern nur in einer neuen Spalte anzeigen. Diese "Anzeigespalten" erhalten keinen Namen.

Um die Binärdarstellung des 1. Faktor gleichzeitig zu erzeugen ist die modulo-Funktion direkt geeignet.

mod$(413,2)$ • 1

mod$($**halb**$(413),2)$ • 0

1.2

	A	B zw..	C erst...	D dual	E halbe	F sp2	G nimm	H	I	J	K
•											
1	0	1	1	1	413	12	12	erst	zweit		
2	1	2	0	0	206	24	0	413	12		
3	2	4	4	1	103	48	48		produkt		
4	3	8	8	1	51	96	96	ägypt	4956		
5	4	16	16	1	25	192	192		summe(ni...		
6	5	32	0	0	12	384	0	heute	4956		
7	6	64	0	0	6	768	0	413	summe(c)		
8	7	128	128	1	3	1536	1536				
9	8	256	256	1	1	3072	3072				
10	9	512	0	0	0	6144	0				
11	10	1024	0	0	0	12288	0				
12	11	2048	0	0	0	24576	0				
13	12	4096	0	0	0	49152	0				
14	13	8192	0	0	0	98304	0				
15	14	163...	0	0	0	196608	0				
16	15	327...	0	0	0	393216	0				

K9

1.3

Stammbruchzerlegung

Stammbruchzerlegung..Die Ägypter haben (fast nur) mit Stammbrüchen, also solchen mit Zähler 1, gerechnet. Sie hatten die Tabelle des Ahmed, der für etliche Brüche Stammbruchzerlegungen angab.

```
Define stammz(a,n)=Func
    Local k,z
    For k,2,20·n+1
      z:=a·k−n
      If z≥0 and gcd(|z|,k·n)=|z| Then
        Return [1/k  z/(k·n)]
      EndIf
    EndFor
    Return "nicht gefunden "
  EndFunc
```

• *Fertig* **stammz**$(4,7)$ • $\begin{bmatrix}\frac{1}{2} & \frac{1}{14}\end{bmatrix}$

stammz$(10,91)$ • $\begin{bmatrix}\frac{1}{14} & \frac{1}{26}\end{bmatrix}$ Probe $\frac{1}{14}+\frac{1}{26}$ • $\frac{10}{91}$ **stammz**$(5,7)$ • nicht gefunden

seq$($**stammz**$(2,n),n,1,10)$ • Fehler: Liste oder Matrix ungültig

stammz$(2,3)$ • $\begin{bmatrix}\frac{1}{2} & \frac{1}{6}\end{bmatrix}$ **stammz**$(2,5)$ • $\begin{bmatrix}\frac{1}{3} & \frac{1}{15}\end{bmatrix}$ **stammz**$(2,7)$ • $\begin{bmatrix}\frac{1}{4} & \frac{1}{28}\end{bmatrix}$ **stammz**$(2,9)$ • $\begin{bmatrix}\frac{1}{5} & \frac{1}{45}\end{bmatrix}$

2.1

Einige Beispiele wie die Tabelle des Ahmed

stammz$(2,11)$ • $\begin{bmatrix}\frac{1}{6} & \frac{1}{66}\end{bmatrix}$ **stammz**$(2,13)$ • $\begin{bmatrix}\frac{1}{7} & \frac{1}{91}\end{bmatrix}$ **stammz**$(2,15)$ • $\begin{bmatrix}\frac{1}{8} & \frac{1}{120}\end{bmatrix}$

seq$($**stammz**$(2,n),n,1,10)$ • Fehler: Liste oder Matrix ungültig   (warum nicht?)

stammz$(2,3)$ • $\begin{bmatrix}\frac{1}{2} & \frac{1}{6}\end{bmatrix}$ **stammz**$(2,5)$ • $\begin{bmatrix}\frac{1}{3} & \frac{1}{15}\end{bmatrix}$ **stammz**$(2,7)$ • $\begin{bmatrix}\frac{1}{4} & \frac{1}{28}\end{bmatrix}$ **stammz**$(2,9)$ • $\begin{bmatrix}\frac{1}{5} & \frac{1}{45}\end{bmatrix}$

stammz$(2,11)$ • $\begin{bmatrix}\frac{1}{6} & \frac{1}{66}\end{bmatrix}$ **stammz**$(2,13)$ • $\begin{bmatrix}\frac{1}{7} & \frac{1}{91}\end{bmatrix}$ **stammz**$(2,15)$ • $\begin{bmatrix}\frac{1}{8} & \frac{1}{120}\end{bmatrix}$

Bei kürzbaren Brüchen findet das Programm nicht die einfachste Version.

stammz$(2,4)$ • $\begin{bmatrix}\frac{1}{3} & \frac{1}{6}\end{bmatrix}$ **stammz**$(2,6)$ • $\begin{bmatrix}\frac{1}{4} & \frac{1}{12}\end{bmatrix}$ **stammz**$(2,8)$ • $\begin{bmatrix}\frac{1}{5} & \frac{1}{20}\end{bmatrix}$ **stammz**$(2,10)$ • $\begin{bmatrix}\frac{1}{6} & \frac{1}{30}\end{bmatrix}$

stammz$(2,12)$ • $\begin{bmatrix}\frac{1}{7} & \frac{1}{42}\end{bmatrix}$ **stammz**$(2,14)$ • $\begin{bmatrix}\frac{1}{8} & \frac{1}{56}\end{bmatrix}$ **stammz**$(2,16)$ • $\begin{bmatrix}\frac{1}{9} & \frac{1}{72}\end{bmatrix}$

Darauf kommt es mir jetzt aber nicht an. Die findet man ja auch im Kopf.

stam$(2,4)$ • $\left\{\frac{1}{2}\right\}$ **stammz**$(5,7)$ • nicht gefunden   aber **stam**$(5,7)$ • $\left\{\frac{1}{2},\frac{1}{5},\frac{1}{70}\right\}$

Das Programm stam(a,n) von der nächsten Seite hat diese Mängel nicht.

2.2

Andere Methode, von der bewiesen ist, dass sie immer endet.

```
Define stam(a,n)=Func
    Local k,li,r,z
    li:={[]}: r:=a/n:
    While r>0
      k:=ceiling(1/r): li:=augment(li,{1/k}):   r:=r−1/k:
    EndWhile
    Return li
  EndFunc
```

• *Fertig*

stam$(5,7)$ • $\left\{\frac{1}{2},\frac{1}{5},\frac{1}{70}\right\}$ **zert**$:=$**stam**$(10,91)$ • $\left\{\frac{1}{10},\frac{1}{102},\frac{1}{11603},\frac{1}{269247615}\right\}$

factor$($**zert**$)$ • $\left\{\frac{1}{2\cdot 5},\frac{1}{2\cdot 3\cdot 17},\frac{1}{41\cdot 283},\frac{1}{3\cdot 5\cdot 7\cdot 13\cdot 17\cdot 41\cdot 283}\right\}$ nun wird die Zerlegung von $\frac{10}{91}$ viel länger, dafür klappt aber eine Zerlegung von $\frac{5}{7}$.

stam$(2,7)$ • $\left\{\frac{1}{4},\frac{1}{28}\right\}$ **stammz**$(2,7)$ • $\begin{bmatrix}\frac{1}{4} & \frac{1}{28}\end{bmatrix}$ Hier liefern beide Programme dasselbe.

2.3