

Vielleicht hätte hier auch POLYA ein CAS eingesetzt

Dieser Artikel befasst sich vordergründig mit der faszinierenden Aufgabe, die POLYA in seinem berühmten Film vor rund einem halben Jahrhundert mit Collegestudenten gelöst hat:

In wie viele Teile wird der Raum durch n Ebenen im Maximum zerlegt?

Ich würde mich hüten, an Polya's paradigmatischen Lektion irgend etwas verbessern zu wollen. Mir geht es darum, sein Vermächtnis aufzufrischen und den bewusst weggelassenen Beweis für die erarbeitete Vermutung zu erbringen und ausserdem zu zeigen, dass sich die Aufgabe über das Bekannte hinaus dazu eignet, eine Reihe von fundamentalen Fragestellungen und Konzepten aus der Mathematik und Informatik anzuschneiden, insbesondere wenn ein CAS eingesetzt wird.

Es zahlt sich in verschiedener Hinsicht aus, wenn man zuerst das analoge und leichtere Problem in der Ebene löst.

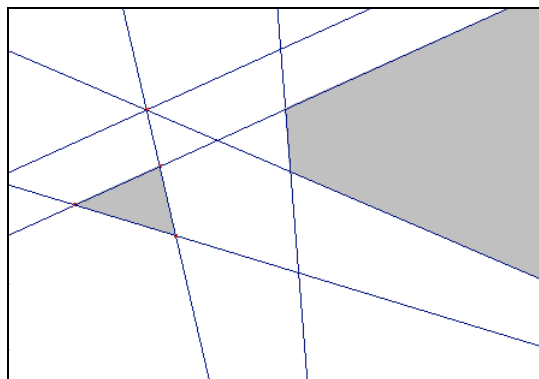
Die analoge Aufgabe in der Dimension 2:

In der Ebene sind n Geraden gezeichnet. In wie viele Gebiete wird sie im Maximum zerlegt?

Eine Herausforderung

Man kann m. E. nicht davon ausgehen, dass die heutigen Schüler und Schülerinnen die Aufgabe ohne weiteres interessant finden. Durch eine geeignete Präsentation des Problems kann man jedoch möglicherweise ihren Ehrgeiz herausfordern.

In der nebenstehenden Zeichnung wird beispielsweise zunächst klar gemacht, dass es endliche und unendliche Gebiete gibt. Und dass es ganz einfach ist, die Geraden zu zählen, aber schon schwieriger, die Anzahl der Gebiete zu erfassen, da einige ja ausserhalb des Rahmens liegen.



Ausserdem kann man anhand dieser Skizze auch gerade klären, dass für die Fragestellung

- parallele Geraden und
- mehr als zwei Geraden durch einen Punkt ausgeschlossen werden müssen.

Übersicht gewinnen

Es liegt auf der Hand, zunächst eine Folge von Bildern mit $n = 1, 2, \dots$ Geraden zu zeichnen und die Anzahl g_n der Gebiete zu zählen.

n	1	2	3	4
n Geraden				
g_n	2	4	7	11

Eine Vermutung durch unvollständige Induktion

Es ist damit zu rechnen, dass die Schüler entdecken, dass der Anfang einer arithmetischen Folge 2. Ordnung vorliegt, auch wenn dieses Thema vorher nicht behandelt wurde. Sie erinnern sich wahrscheinlich an die einfachste arithmetische Reihe:

$$1 + 2 + 3 + 4 + \dots$$

und schliessen, wie wir sehen werden, richtig auf

$$g_n = 1 + (1 + 2 + \dots + n) = 1 + \frac{n(n+1)}{2}.$$

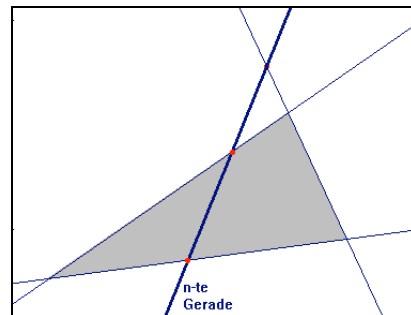
Das gäbe dem Lehrer die Gelegenheit, darauf hinzuweisen, dass es sich gewiss um eine interessante und wertvolle Beobachtung handelt, die allerdings aus der Sicht der Mathematik als unsicher und unbewiesen angesehen werden muss, selbst dann, wenn sie in Dutzenden von weiteren Fällen zutreffen sollte. Man nennt das Schliessen aufgrund weniger Glieder der Folge auf alle Folgenglieder „unvollständige Induktion“. Unvollständige Induktion ist in vielen Wissensgebieten, u.a. in der Physik die (notwendige) Regel, in der Mathematik aber nicht akzeptiert.

Lösung durch die rekursive Definition der Folge g_n

Nur die aus der Geometrie abgeleitete rekursive Definition von g_n wird den mathematischen Anforderungen gerecht.

Die n -te gezeichnete Gerade (in nebenstehender Figur fett) wird von den $n-1$ vorher gezeichneten Geraden in maximal $n-1$ Punkten geschnitten und damit in n Strecken oder Halbgeraden zerlegt.

Jeder dieser n Geradenteile zerlegt ein bereits vorhandenes Gebiet in zwei Teile (z.B. das getönte Dreieck in ein Drei- und ein Viereck). Die bisherigen g_{n-1} Gebiete bleiben also teilweise oder ganz erhalten und es kommen n neue dazu.



Diese Überlegung bestätigt die obige Vermutung und führt auf die folgende *rekursive Definition* der Folge g_n :

$$g_1 = 2$$

$$g_n = g_{n-1} + n \quad \text{für } n > 1$$

Berechnung von g_n durch einen rekursiven Prozess

Die obige Rekursionsformel kann in einem CAS praktisch eins zu eins programmiert werden. Beim TI-89/voyage lautet der Befehl beispielsweise:

```
DEFINE G(N) = WHEN(N > 1, G(N-1) + N, 2)
```

Der Computer bestätigt die Eingabe in sehr übersichtlicher Form:

$$g(n) = \begin{cases} 2, & n = 1 \\ n + g(n-1), & \text{else} \end{cases}$$

Die Kontrolle der Formel anhand der oben gezeichneten Fälle fällt zufriedenstellend aus:

$$g(1) = 2$$

$$g(2) = 4$$

$$g(3) = 7$$

$$g(4) = 11$$

Wie entwickelt sich die Folge?

$$g(10) = 56$$

$$G(20) = 211$$

$$G(50) = \text{ERROR MEMORY}$$

Eine böse Überraschung! Kann man sie verstehen?

Meines Erachtens kommt es nicht darauf an, dass man *genau* (mit Stapelregistern etc) erklärt, was im Computer passiert. Die wesentliche Ursache der Fehlermeldung erkennt man, wenn man sich überlegt, was beispielsweise bei der Berechnung von g_6 durch das obige Programm abläuft (wenn der Programminterpretierer jeweils versucht, zuerst die innerste Klammer auszuwerten).

$$\begin{aligned} G(6) &= G(5) + 6 \\ &= (G(4) + 5) + 6 \\ &= ((G(3) + 4) + 5) + 6 \\ &= (((G(2) + 3) + 4) + 5) + 6 \\ &= (((G(1) + 2) + 3) + 4) + 5) + 6 \\ &= (((2 + 2)) + 3) + 4) + 5) + 6 \\ &= (((4 + 3) + 4) + 5) + 6 \\ &= ((7 + 4) + 5) + 6 \\ &= (11 + 5) + 6 \\ &= 16 + 6 \\ &= 22 \end{aligned}$$

Die Darstellung macht eindrücklich sichtbar, wie der Verarbeitungsprozess auf eine immer länger werdende Kette von *nicht ausrechenbaren* Klammern führt, die „intern notiert“ werden müssen, wozu Speicherplatz –und auch Zeit– benötigt wird. Wenn der dafür vorgesehene Speicherumfang überschritten wird, meldet sich der Computer eben mit der Fehlermeldung „Error Memory“. (Genauere Erklärungen zum Einsatz von Stapelregistern findet man im Leitprogramm [2].)

Es ist m. E. wichtig, zu betonen, dass dieses Ereignis grundsätzlich auf jedem –noch so grossen– Computer durch ein genügend kompliziertes Rekursionsprogramm erzeugt werden kann.

Berechnung von g_n mit einem iterativen Prozess

Die Idee der *iterativen* Berechnung ist klar: g_1 ist gegeben; mit der Rekursionsformel können der Reihe nach g_2, g_3, \dots berechnet werden.

$$\begin{aligned} g_1 &= 2 \\ g_2 &= g_1 + 2 = 4 \\ g_3 &= g_2 + 3 = 7 \\ &\dots \end{aligned}$$

Genau so rechnet der TI-89/voyage, wenn man die Folge g_n unter

MODE GRAPH.....SEQUENCE

im $\boxed{Y=}$ -Menü programmiert:

$$\begin{aligned} U1 &= U1(N - 1) + N \\ U11 &= 2 \end{aligned}$$

Im HOME-Menü erhalten wir so problemlos die Werte von Gliedern mit noch so grosser Nummer:

$$U_1(100) = 5051.$$

Aufgrund des obigen Rechenschemas ist unmittelbar klar, dass der *iterative* Rechenprozess im Computer keinen zusätzlichen Speicherplatz beansprucht und dass deshalb die Berechnung von g_n bei grossem n nur eine Frage der Zeit ist.

Vergleich Rekursion / Iteration

Die *rekursiv definierte Folge* g_n wurde oben mit zwei verschiedenen Programmen berechnet, die zwar für ein gegebenes n praktisch gleich viele Operationen –proportional zu n - erfordern, aber im ersten Fall zu einem erheblichen zusätzlichen Speicherbedarf führen. Man bezeichnet die erste Berechnungsart als einen (*linear*) *rekursiven Prozess* und die zweite als einen (*linear*) *iterativen Prozess* [1]. In [1] wird betont, dass man „rekursive Definition“ und „rekursiven Prozess“ auseinanderhalten muss.

Das Umschreiben von rekursiven Programmen in iterative ist aus den geschilderten Gründen ein Standardproblem der Informatik. Bei den rekursiv definierten Folgen ist das, wie gesagt, trivialerweise möglich. Für viele wichtige Probleme kennt man aber leider nur rekursive Lösungen.

Im ersten Artikel der ETH-Fallstudie „Rekursion“ [3] wird das Spannungsverhältnis zwischen Rekursion und Iteration als das Yin und Yang der Informatik bezeichnet.

Vermutung für eine explizite Formel für g_n

Auch wenn die Folge g_n iterativ berechnet wird, benötigt der Computer dafür eine beachtliche Zeit, weil viele Operationen durchzuführen sind. U.a. deshalb entsteht der Wunsch nach einer sog. *expliziten Formel*, einer Funktion $x \rightarrow f(x)$ mit der Eigenschaft:

$$n \rightarrow g_n = f(n) \quad \text{für } n \in \mathbb{N},$$

welche also ermöglicht, g_n für beliebig grosses n „*instantan*“ zu berechnen.

Es gibt verschiedene Strategien, die Formel f zu erraten. Oft erkennt man f , wenn die Glieder g_n faktorisiert werden. Hier -wir gehen davon aus, dass arithmetischen Folgen höherer Ordnung nicht behandelt wurden- gewinnt man f , indem die durch die Rekursionsformel induzierte „Geschichte“ der Glieder g_n ausgeschrieben wird:

$$g_1 = 2$$

$$g_2 = 2 + 2$$

$$g_3 = 2 + 2 + 3$$

$$g_4 = 2 + 2 + 3 + 4$$

Die Struktur dieser vier Terme führt auf die bereits eingangs des Artikel geäusserte Vermutung:

$$\begin{aligned} g_n &= 2 + 2 + 3 + \dots + n \\ &= 1 + 1 + 2 + 3 + \dots + n \\ &= 1 + \frac{n+1}{2}n, \end{aligned}$$

und damit auf die Funktion:

$$f(x) = 1 + \frac{x+1}{2}x.$$

Sollte die Formel für die Summe der ersten n natürlichen Zahlen fehlen, so hilft das CAS. Die Eingabe

$$\Sigma(1, 1, 1, N)$$

quittiert der TI-89/voyage nämlich mit:

$$\sum_{i=1}^n i = \frac{n \cdot (n + 1)}{2}$$

Stellenwert expliziter Formeln

An dieser Stelle sollte man m.E. darauf hinweisen, dass das Finden einer Formel f sehr schwierig sein kann oder sogar unmöglich ist.

Beispielsweise wurden die Binetformeln für die Fibonacci-Folge erst im 19. Jh. entdeckt. Aus dieser Zeit stammt wahrscheinlich auch der Trick, die Formeln mit Hilfe einer erzeugenden Funktion zu berechnen.

Oder für die Heronfolge

$$x_1 = 1, \quad x_{n+1} = \frac{x_n + \frac{2}{x_n}}{2} \quad (n \in \mathbb{N})$$

die $\sqrt{2}$ approximiert, ist eine explizite Formel nicht bekannt (und steht gar nicht zur Diskussion, weil die Folge ja rasch konvergiert).

Beweis der expliziten Formel (evtl. mit CAS)

Nach der obigen Herleitung sind die SchülerInnen schlecht motiviert, die Vermutung für f auch noch zu beweisen, weil sie „sicher“ sind, dass sie stimmt. Wäre die Vermutung jedoch –wie so oft– aus dem „hohlen Bauch heraus“ entstanden, wäre ein Beweis wohl unerlässlich.

Im Vordergrund steht die Methode der vollständigen Induktion, die allerdings aus Zeitgründen nicht mehr an allen Schulen thematisiert wird. Man kommt im vorliegenden Fall jedoch mit einer etwas einfacheren, unmittelbar einleuchtenden Argumentation zum Ziel:

Fehler!

Die Zahlenfolge $f(1), f(2), \dots$ ist identisch mit g_1, g_2, \dots , wenn

- die Startwerte gleich sind, d.h. $g_1 = f(1)$ und
- wenn sie die gleiche Rekursionsformel erfüllt wie g_n , nämlich $f(n) = f(n-1) + n$.

Am bequemsten beweist man dies mit einem CAS, indem man zuerst f definiert:

$$\text{DEFINE } F(X) = 1 + \frac{X+1}{2} X$$

und hernach die zwei obigen Bedingungen (erfolgreich!) prüft:

- $F(1) = G(1)$ TRUE
- $F(N) = F(N-1) + N$ TRUE

In diesem einfachen Falle hätte man natürlich (fast) allen SchülerInnen den Nachweis durch Handrechnung zumuten dürfen:

- $f(1) = 1 + \frac{1+1}{2} \cdot 1 = 1 + 1 = 2$
- $f(n-1) + n = 1 + \frac{n-1+1}{2} \cdot (n-1) + n = \frac{n^2}{2} + \text{Fehler!} + 1$

$$f(n) = 1 + \frac{n+1}{2}n = \frac{n^2}{2} + \text{Fehler!} + 1$$

Die beiden rechten Seiten der Gleichungen sind gleich, also auch die linken.

Vorläufige Zusammenfassung

Bei der Lösung des zweidimensionalen Falls sind wir neben dem beispielhaften mathematischen Tun ganz natürlich auf eine unerwartete Fülle von grundsätzlichen Themen der Mathematik und Informatik zu sprechen gekommen. Stichworte sind:

- Rekursive und explizite Definition einer Folge
- Rekursive und iterative Programme zur Berechnung einer Folge
- Unvollständige und (vereinfachte) vollständige Induktion

Mögliche Anschluss- oder Prüfungsaufgaben sind:

- Wie viele *endliche* Gebiete gibt es im Maximum bei n Geraden?
- Wie viele Gebiete gibt es im Maximum bei n Kreisen?

Auf das eigentliche Problem sind wir scheinbar noch überhaupt nicht eingegangen. Wie sich jedoch zeigen wird, sind die Analogien in jeder Hinsicht derart stark, dass man die vollständige Lösung mit gutem Gewissen der Klasse als eine Projektarbeit überlassen könnte.

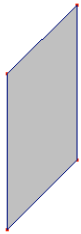
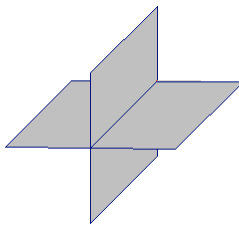
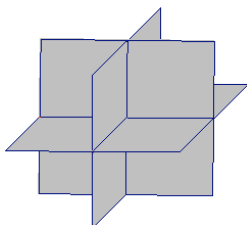
Das Hauptproblem:

In wie viele Gebiete wird der Raum im Maximum durch n Ebenen zerlegt?

Übersicht gewinnen

Es liegt auch hier auf der Hand, zunächst eine Folge von Bildern mit n = 1, 2, ... Ebenen zu zeichnen und die Anzahl G_n der Gebiete zu zählen. Dabei wird G_n maximal, wenn keine zwei Ebenen parallel sind und keine drei Ebenen eine gemeinsame Schnittgerade haben.

Einfachheitshalber wählen wir für unsere Skizzen die Koordinatenebenen eines dreidimensionalen orthogonalen Koordinatensystems, die je nach Anzahl den Raum in zwei Halbräume, vier Quadranten bzw. acht Oktanten zerlegen.

n	1	2	3	4
n Ebenen				Es ist für den unbelasteten Anfänger nicht ganz einfach, ohne fremde Hilfe eine vierte Ebene einzuführen und die Anzahl Raumgebiete zu zählen.
G _n	2	4	8	

Die rekursive Definition von G_n

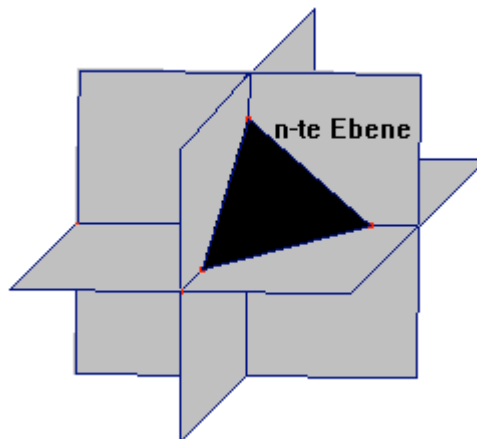
Inzwischen sind wir so vorsichtig, dass wir der naheliegende Vermutung

$$G_n = 2^n$$

sehr misstrauen. Wir wissen sowieso, dass wir die rekursive Definition von G_n aus der Geometrie herleiten müssen.

Wir denken uns $n-1$ Ebenen bereits vorhanden und überlegen uns, was passiert, wenn wir die n -te Ebene hinzufügen (in nebenstehender Skizze als schwarzes Dreieck angedeutet):

- Sie schneidet die $n-1$ vorhandenen Ebenen in $n-1$ Schnittgeraden.
- Diese $n-1$ Schnittgeraden definieren in der n -ten Ebene g_{n-1} Gebiete (in nebenstehender Zeichnung z.B. das Dreieck im 1. Oktanten).
- Jedes dieser Gebiete zerlegt einen bereits vorhandenen Raumteil in zwei Teile (beispielsweise trennt das schwarze Dreieck dem 1. Oktanten ein Tetraeder ab).



Hieraus ergibt sich folgende rekursive Definition für G_n :

$$G_1 = 2$$

$$G_n = G_{n-1} + g_{n-1} \quad \text{für } n > 1$$

Wenn wir die explizite Formel f für die Folge (g_n) benutzen, ergibt sich für $n > 1$:

$$G_n = G_{n-1} + f(n-1) \quad \text{oder} \quad G_n = G_{n-1} + 1 + \frac{n \cdot (n-1)}{2}.$$

Erste Folgerungen

In der obigen Übersicht waren wir mit G_4 überfordert. Die Rekursionsformel liefert nun:

$$G_4 = G_3 + 1 + \frac{4 \cdot 3}{2} = 8 + 1 + 6 = 15,$$

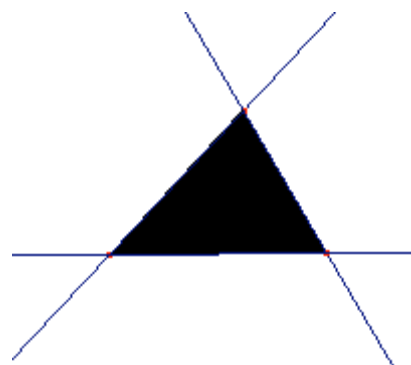
d.h. zu den acht, nun teilweise angeschnittenen Oktanten sind sieben neue Raumgebiete dazugekommen.

Mit Hilfe der Überlegungen bei der Herleitung können wir uns diese 15 Raumteile auch ansatzweise vorstellen.

Die nebenstehende Skizze zeigt uns die drei Schnittgeraden in der 4. Ebene, die das schwarze Dreieck und drum herum sechs unendliche Gebiete definieren.

Das Dreieck ist die Grundfläche des Tetraeders mit Spitze im Koordinatenursprung, der im 1. Oktanten abgeschnitten wird.

Die sechs unendlichen Ebenengebiete zerlegen die Oktanten 2 bis 7 in je zwei unendliche Raumteile. Nur der 8. Oktant bleibt unzerlegt.



Eine Vermutung für die explizite Formel von G_n

Zur Abwechslung probieren wir einen neuen Ansatz aus. Grundlage sind die Ergebnisse für die analogen Probleme in den Dimensionen 1 und 2:

- Eine Gerade wird durch n Punkte in

$$\gamma_n = 1 + n$$

Streckenteile zerlegt; γ_n ist ein Polynom 1. Grades.

- Eine Ebene wird, wie wir gesehen haben, durch n Geraden in

$$g_n = 1 + \frac{n(n+1)}{2}$$

Ebenengebiete zerlegt; g_n ist ein Polynom 2. Grades, wie γ_n mit dem konstanten Glied 1.

- Wir vermuten deshalb, dass G_n durch ein solches Polynom 3. Grades dargestellt werden kann:

$$G_n = F(n), \quad \text{wobei } F(x) = 1 + a \cdot x^3 + b \cdot x^2 + c \cdot x \quad (a, b, c \in \mathbb{Q}).$$

Zur Berechnung von a, b, c stützen wir uns auf die drei Werte $G_1 = 2$, $G_2 = 4$, $G_3 = 8$. Die drei Unbekannten müssen also das Gleichungssystem:

$$F(1) = 2$$

$$F(2) = 4$$

$$F(3) = 8$$

erfüllen. Man kann dieses System so eins zu eins in das CAS eingeben. Auf die Eingabe (wir schreiben ff statt F):

DEFINE FF(X) = 1 + A*X³ + B*X² + C*X

SOLVE (FF(1) = 2 AND FF(2) = 4 AND FF(3) = 8, {A, B, C})

antwortet beispielsweise der TI voyage mit:

A = 1/6 AND B = 0 AND C = 5/6

d.h. das postulierte Polynom F müsste folgende Gleichung haben:

$$F(x) = 1 + \frac{x^3 + 5x}{6}.$$

Beweis mit (vereinfachter) vollständiger Induktion

Ohne Beweis ist unsere Vermutung wertlos, denn selbst wenn es keine Funktion F für G_n gäbe, hätte das obige System diese Lösung.

Dass für beliebiges $n \in \mathbb{N}$ gilt:

$$G_n = F(n), \quad \text{wenn } F(x) = 1 + \frac{x^3 + 5x}{6},$$

beweisen wir analog zum zweidimensionalen Fall mit Hilfe eines CAS:

- Wir definieren F: **DEFINE FF(X) = 1 + (X³ + 5X)/6**
- Ebenso G_n : **DEFINE GG(N) = WHEN (N>1, GG(N-1) + F(N-1), 2)**
- Nun beweisen wir, dass F den gleichen Startwert hat wie G_n , d.h. $F(1) = G_1$:

$$\mathbf{FF(1) = GG(1) \quad TRUE}$$

- Und wir zeigen, dass F die gleiche Rekursionsformel wie G_n erfüllt, d.h. $F(n) = F(n-1) + f(n-1)$:

$$\mathbf{FF(N) = FF(N-1) + F(N-1) \quad TRUE}$$

Die Mehrzahl der SchülerInnen kann m. E. diesen übersichtlichen Beweis nachvollziehen und verstehen. Der Nachweis mit Hilfe einer Handrechnung würde wohl wenig zum Verständnis beitragen, im Gegenteil ist zu befürchten, dass die bereits recht anspruchsvollen Umformungen einige überfordern würde.

Alternative Herleitung von F

Aufgrund der rekursiven Definition

$$G_1 = 2$$

$$G_n = G_{n-1} + g_{n-1}, \quad \text{für } n > 1$$

kann F auch wie folgt mit Hilfe eines CAS berechnet werden:

$$G_1 = 2$$

$$G_2 = G_1 + g_1$$

$$G_3 = G_1 + g_1 + g_2$$

Offenbar gilt:

$$G_n = 2 + g_1 + g_2 + \dots + g_{n-1}$$

oder

$$G_n = 2 + f(1) + f(2) + \dots + f(n-1), \quad \text{wobei } f(x) = 1 + \frac{x \cdot (x+1)}{2}$$

Der TI voyage quittiert die Eingabe

$$2 + \Sigma(F(1), 1, 1, N-1)$$

mit dem oben bewiesenen Ergebnis für F:

$$2 + \sum_{i=1}^{n-1} f(i) \qquad \frac{N^3}{6} + \frac{5N}{6} + 1$$

Bei dieser Herleitung erübrigt sich m.E. ein zusätzlicher Beweis. Wahrscheinlich würden viele SchülerInnen diesen Weg sogar selbständig finden.

Binomialkoeffizienten in F

In den Polynomen für γ_n , g_n und G_n kommen die Nenner 1, 2 und 6 vor, d.h. die Fakultäten 1!, 2! und 3!. Deshalb ist es naheliegend, zu prüfen, ob die Polynome durch Binomialkoeffizienten ausgedrückt werden können. In der Tat gelten:

$$\gamma_n = 1 + n = (n;0) + (n;1)$$

$$g_n = 1 + \frac{n(n+1)}{2} = (n;0) + \binom{n+1}{2} = (n;0) + (n;1) + (n;2)$$

Wiederum mit dem CAS erkennt man, dass die auf der Hand liegende Vermutung

$$G_n = (n;0) + (n;1) + (n;2) + (n;3)$$

zutrifft. Der TI voyage bestätigt die Eingabe

$$\Sigma(\text{NCR}(N,1), 1, 0, 3)$$

mit

$$\sum_{i=0}^3 nCr(\eta, i) \qquad \frac{N^3}{6} + \frac{5N}{6} + 1$$

Man kann also die Anzahlfunktionen für die Dimensionen $d = 1, 2, 3$ in einer einzigen zusammenfassen:

Fehler!

Der Autor:

Otto M. Keiser
Hinterbergstrasse 88
CH-8044 Zürich
omkeiser@smile.ch

Literatur:

- [1] *H. Abelson, G.J. Sussman*: Structure and Interpretation of Computer Programs, 1985, The MIT Press
- [2] *A. Alder u.a.*: Rekursives Programmieren, Ein Leitprogramm in Informatik (für den TI 92 bearbeitet von Otto M. Keiser), herunterladbar auf www.educETH.ch / Informatik / Leitprogramme
- [3] *A. Gächter*: ETH-Fallstudie Rekursion, 1991, Orell Füssli Verlag
(erst nach Verfassen des Artikels zufällig gefunden)