Lasse Berglund

Rekursiva betraktelser

Denna artikel bygger vidare på övningar som presenterades i artikeln *Bestämma π med Python* i Nämnaren 2021:1. I en blandning av allvar och lek får vi ta del av rekursiva betraktelser över några välkända summor.

Talen i talföljden 1, 3, 6, 10, 15, 21 ... är de så kallade triangeltalen. Figuren nedan visar orsaken till denna benämning. Dessa tal kan helt enkelt arrangeras så att de bildar trianglar.

Från figuren framgår också principen för hur de bildas. Om vi kallar talen för T, kan vi nu formulera denna princip, explicit såväl som implicit.

Explicit	Implicit
$T_1 = 1$	$T_1 = 1$
$T_2 = 1+2$	$T_2 = T_1+2$
$T_3 = 1+2+3$	$T_3 = T_2+3$
$T_4 = 1+2+3+4$	$T_4 = T_3+4$
...	...
$T_n = 1+2+3+4+\ldots+n$	$T_n = T_{n-1}+n$ med basfallet $T_1 = 1$

Strax ska vi ta hjälp av det välbekanta samband som den åttaårige *Gauss* fann. Men utan att känna till detta samband, kan vi ändå snabbt räkna ut till exempel T_{679} om vi skriver några rader kod enligt den implicita idén ovan, det vill säga rekursivt. Det programspråk vi kommer att använda på dessa sidor är Python.

Triangeltalen

```
n = int(input("Ange vilket triangeltal: "))
def T(n):
    if n == 1:
        return 1
    else:
        return T(n – 1) + n
print T(n)
```

```
#  Användaren får ange sitt triangeltal.
#  En funktion T(n) definieras.
#  if-satsen anger det viktiga basfallet.


#  Rekursionen går igång med sitt självanrop.
#  Det aktuella triangeltalet skrivs ut.
```

```
Ange vilket triangeltal:  679
230860
```

Av koden framgår att det 679:e triangeltalet har värdet 230860. I Python används symbolen = = för en likhet, symbolen = är en tilldelning.

Historisk anekdot

Vad Gauss gjorde den där gången i klassrummet i slutet av 1700-talet, var att han beräknade summan av de första hundra heltalen. Han hade upptäckt den struktur som fanns i uppgiften, och kunde därför inom någon halvminut ange sitt svar till 5050. Den läsare som vill kontrollera pojkens uträkning kan skriva in koden i sin dator och sen köra T(100).

Gauss, 8 år

Om man ska bygga pyramider

Tar vi så fram det samband som Gauss fann:

$$T_n = 1+2+3+4+\ldots+n = n(n+1)/2$$

får vi så klart samma resultatet som ovan: $T_{679} = 679 \cdot 680/2 = 230860$. Men nu vill vi gå vidare och inte bara beräkna dessa triangeltal tagna var för sig, utan summan av dem, det vill säga $S_n = 1+3+6+10+15+\ldots+T_n$. Så med Gauss formel och med den rekursiva approachen, får vi än en gång enkelt och utan krusiduller:

$S_n = S_{n-1} + T_n$ det vill säga:

$S_n = S_{n-1} + n(n+1)/2$ med basfallet $S_1 = 1$

Pyramidtalen

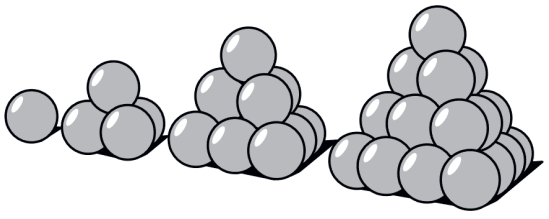
```
n = int(input("Ange antalet triangeltal: "))
def S(n):
    if n == 1:
        return  1
    else:
        return  S(n – 1) + n*(n+1)/2
print S(n)
```

\# Summan anropar sig själv, dvs S(n) anropar S(n – 1).

```
Ange antalet triangeltal:  457
16011909
```

Då triangeltalen har en formation som är lämplig om man vill stapla kulor, skulle en tolkning av resultatet i figuren ovan kunna vara, att ska man bygga sig en kulpyramid som ska vara 457 våningar hög, så kommer det att gå åt drygt 16 miljoner kulor.

De fyra pyramiderna i figuren nedan, består av 35 kulor sammanlagt. Den intresserade kan visa att de första 600 pyramiderna består av sammanlagt 5 454 165 150 kulor.



Triangeltalens reciproker

Vi fortsätter med triangeltalen men tar nu den multiplikativa inversen på var term och ger oss därmed i kast med summan av deras så kallade reciproker. Lite räkning med pennan får inleda:

$$1 + 1/3 + 1/6 + 1/10 = 48/30 = 1{,}6$$

- Kan vi få en maskin att beräkna även denna summa snabbt och enkelt?
- Kan summan bli hur stor som helst?
- Kan eventuella resultat bevisas?

Vi tar frågorna som de kommer, kallar summan för R och utnyttjar än en gång Gauss samband: $T_n = n(n+1)/2$

$$R_n = 1 + \frac{1}{3} + \frac{1}{6} + \frac{1}{10} + \ldots + \frac{2}{n(n+1)}$$

Då temat för dagen är rekursivitet, följer nu lekande lätt:

$$R_n = R_{n-1} + 2/(n(n+1)) \qquad \text{med basfallet } R_1 = 1$$

Summan av triangeltalens reciproker

```
n = int(input("Ange antal termer: "))
def R(n):
    if n == 1:
        return  1.0
    else:
        return  R(n – 1) +  2/(n*(n+1.0))
print R(n)
```

```
Ange antal termer:  491
1.99593495935
```

Med hjälp av Gauss och lite partialbråksuppdelning ska vi bemöta frustrationen ovan.

$$R_4 = 1 + \frac{1}{3} + \frac{1}{6} + \frac{1}{10} = 1 + \frac{2}{2\cdot 3} + \frac{2}{3\cdot 4} + \frac{2}{4\cdot 5} = 1 + \left(\frac{2}{2} - \frac{2}{3}\right) + \left(\frac{2}{3} - \frac{2}{4}\right) + \left(\frac{2}{4} - \frac{2}{5}\right) = 2 - \frac{2}{5}$$

Det mesta försvann. Bara 1:an och de två yttertermerna blev kvar. Med n stycken termer i stället för 4, får vi:

$$R_n = 2 - \frac{2}{n+1}$$

Då n alltid är positivt blir $2/(n+1)$ alltid positivt. Summan är alltså 2 *minus* ett positivt tal, det vill säga summan av de reciproka triangeltalen kommer aldrig att överstiga 2. Tar vi nu återigen 491 stycken termer, får vi så klart samma resultat som ovan, men denna gång behöver datorn inte göra några 490 stycken anrop då vi nu har sambandet på *sluten form*:

$$R_{491} = 2 - 2/492 = 1{,}99593495935$$

Skulle vi inte nöja oss med triangeltalen utan ta med alla reciproker, $1+1/2+1/3+\ldots+1/n$, får vi den så kallade harmoniska summan och då är plötsligt inte 2 någon övre gräns. Tvärtom höll jag på att säga: den kan bli precis hur stor som helst, även om takten, milt utryckt, är låg. Läs mer om harmoniska serier i artikeln *Bortom vardagen*.

Trevliga komplement

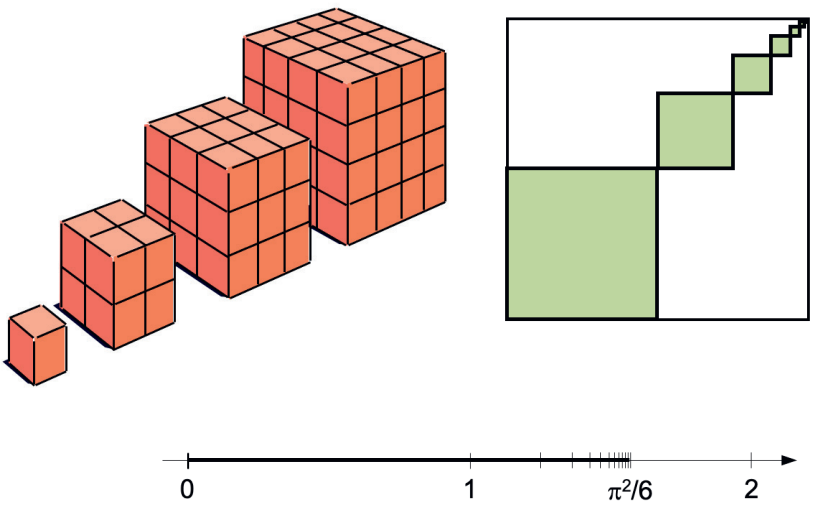
Den intresserade eleven inser snart att dessa summor så klart inte behöver handla om just triangeltal eller reciproker. Då det ibland finns en del fina tolkningar av vissa summor – precis som i fallet med pyramiderna – så avslutar vi dessa rader med några figurer som tillsammans med sina rekursiva formuleringar kompletterar varandra på ett trevligt sätt.

$$S_n = S_{n-1} + n^3 \qquad S_1 = 1 \qquad n = 2, 3, 4$$

$$S_n = S_{n-1} + 1/n^2 \qquad S_1 = 1$$

$$S_n = S_{n-1} + 1/4^n \qquad S_1 = 1/4$$

För den intresserade: para ihop summorna ovan med figurerna nedan. En summa som består av oändligt många termer kallas för en serie. Hur kan man se att en av serierna ovan har värdet 1/3 och hur kan man kontrollera att en annan har det fantastiska värdet $\pi^2/6$?



LITTERATUR

Berglund, L. (2017). *Strövtåg: Bortom vardagen*. Nämnaren 2017:1.
Berglund, L. (2021). *Bestämma π med Python*. Nämnaren 2021:1.