A combinatorics quiz (solution)

A. Droseltis

19th December 2010

1 Simple solution

We remark, that every possibility to express the connexion can be expressed as an ordered solution of the Diophantine equation

$$x + 2y + 3z = n, \tag{1}$$

where n is the connexion gap. E.g., for $n = 4$ the 7 solutions are:

$$\begin{aligned} 4 &= 1+1+1+1 \\ &= 1+1+2 \\ &= 1+2+1 \\ &= 2+1+1 \\ &= 2+2 \\ &= 1+3 \\ &= 3+1 \end{aligned}$$

Thus we have to compute all the solutions of (1) and then use the multinomial coefficient to compute all permutations of each solution $(x\ y\ z)$, which are

$$\frac{(x+y+z)!}{x!\,y!\,z!}.$$

The following Python code does this.

```
from math import factorial
def mc(x,y,z):
        return factorial(x+y+z) / ( factorial(x) * factorial(y) * factorial(z) )

def generations(n):
        count = 0
        for x in range(n+1):
                for y in range(3):
                        RS = n - x - 2*y
```

```
                        if RS % 3 == 0:
                                break
                while RS >= 0:
                        count += mc(x,y,RS/3)
                        y += 3
                        RS = n - x - 2*y
        return count

print generations(100)
print generations(1000)
```

The answers are:

for 100 generations: 180396380815100901214157639

for 1000 generations: 2758842807766486252615892411656158645133100149652696210351601845036392978912293462801016485671033253921841350537004356434253826361707295202024537559785200706502368152965047761644352316799391470273906561574500883480570560512982435681502330814068718832813973880527601

But calculating the 1000 gap lasts (at my computer) a little more than 1 minute. Can we do better?

2 Sophisticated solution

We remark from above, that we want to count all compositions of an integer n with summands at most 3. The generating function for all compositions of an integer n with summands at most r is[1]

$$C^{\{1,\ldots,r\}}(z) = \frac{1}{1 - z\frac{1-z^r}{1-z}}.$$

For $r = 3$ we have

$$f(z) := C^{\{1,2,3\}}(z) = \frac{1}{1 - z - z^2 - z^3}.$$

The 100th and 1000th coefficients of the McLaurin expansion of f are the solutions. The answer is given in a few seconds with the following file to be loaded from Sage:

```
f = 1 / (1 - x - x^2 - x^3)
t = f.taylor(x, 0, 1000)
print t.coefficient(x^100)
print t.coefficient(x^1000)
```

[1] Flajolet Philippe & Sedgewick Robert: Analytic Combinatorics, Paskebryg, 0th Edition (temporary version) 2006, p. 319.