# Ejercicios 23 y 24

February 4, 2018

Ejercicio 23. Se define la función (n) como la suma de todos los divisores positivos del entero positivo n. Sage dispone de la función sigma que realiza este cálculo. Determina (experimentalmente) el mayor entero positivo tal que la diferenciaCAPÍTULO 7. APROXIMACIÓN 200 e ů n ů log(log(n)) (n), es negativa ( es la constante de Euler).

```
In [8]: n = 1
        res = 0
        while res <= 0:
            res = (e^euler_gamma) * n * log(log(n)) - sigma(n)
            print res.n()
            n += 1
        print n - 1
```

```
-infinity
-4.30557210737007
-3.49748202478000
-4.67296291512818
-1.76207180343745
-5.76769190455912
0.299991024183651
7
```

Definimos una sucesión de números racionales mediante G 1 = 0, G 2 = 1, G n = G n1 + 1. Comprueba (experimentalmente) que la fracción n G n 1 G n2 (n 3). n 2 tiende al número e. 2. Compara la eficiencia de esta manera de aproximar e con la de la forma, más conocida, que utiliza la serie

```
In [16]: def Gn(n):
             if n < 1:
                 return -1
             if n == 1:
                 return 0
             if n == 2:
                 return 1
             else:
                 return Gn(n-1) + Gn(n-2)*(1/(n-2))
```

```
In [26]: for i in xsrange(2,15):
             print i/Gn(i).n(digits=20)
         print "e:", e.n(digits=20)

2.0000000000000000000
3.0000000000000000000
2.6666666666666666667
2.7272727272727272727
2.7169811320754716981
2.7184466019417475728
2.7182633317602642756
2.7182836938934499910
2.7182816576664037376
2.7182818427778273385
2.7182818273518744088
2.7182818285384861664
2.7182818284537281837
e: 2.7182818284590452354


In [27]: def serie(n):
             res = 0
             for i in xsrange(0,n+1):
                 res = res + 1/factorial(i)
             return res

In [28]: %%time
         serie (20).n(digits=20)

CPU times: user 0 ns, sys: 0 ns, total: 0 ns
Wall time: 413 ţs


Out[28]: 2.7182818284590452353

In [30]: %%time
         20/Gn(20).n(digits=20)

CPU times: user 20 ms, sys: 0 ns, total: 20 ms
Wall time: 19.5 ms


Out[30]: 2.7182818284590452352

In [32]: e.n(digits=20)

Out[32]: 2.7182818284590452354
```

Podemos ver que el nuevo método es algo más lento que el metodo más usado. Aún así, en las 20 primeras cifras vemos que es, al igual que el otro método, igual de efectivo.