Fractions irréductibles

Cet exercice ainsi que sa correction est proposé par Philippe Moutou. Il enseigne au lycée Henri IV à Paris.

Exercice

On veut parcourir l'ensemble des rationnels strictement positifs en rangeant les fractions $\frac{a}{b}$ dans l'ordre de la somme $s = a + b$ et, pour une même valeur de s, dans l'ordre de a.
L'idée est d'aller jusqu'à une certaine fraction $Q = \frac{A}{B}$ et d'afficher le nombre N et le pourcentage p des fractions irréductibles inférieures ou égales à Q.
Avec $Q = \frac{2}{5}$, les 17 fractions rangées jusqu'à Q étant : $\frac{1}{1}, \frac{1}{2}, \frac{2}{1}, \frac{1}{3}, \frac{2}{2}, \frac{3}{1}, \frac{1}{4}, \frac{2}{3}, \frac{3}{2}, \frac{4}{1}, \frac{1}{5}, \frac{2}{4}, \frac{3}{3}, \frac{4}{2}, \frac{5}{1}, \frac{1}{6}$ et $\frac{2}{5}$ parmi lesquelles 4 sont simplifiables ($\frac{2}{2}, \frac{2}{4}, \frac{3}{3}, \frac{4}{2}$). On doit obtenir $N = 13$ et $Q = \frac{13}{17}$. Jusque-là, on peut se passer des listes mais on souhaite obtenir l'affichage des fractions irréductibles dans l'ordre numérique croissant.

```
def pgcd(a,b):
  reste=a%b
  if reste==0:
    return b
  else:
    return pgcd(b,reste)

def insere(a,b,L):
  rg,val=0,a/b
  for e in L:
    if e[2]>val:
      break
    else:
      rg=rg+1
  L.insert(rg,[a,b,val])

def denombre(n,d):
  som,tot,rang,irr,red=2,0,0,[],[]
  while som<=n+d:
    num,denom=1,som-1
    while num&lt;som:
      PGCD=pgcd(num,denom)
      if PGCD==1:
        rang=rang+1
```

```
25          insere(num,denom,irr)
26        else:
27          insere(num,denom,red)
28        tot=tot+1
29        if num==n and denom==d:
30          return rang,tot,PGCD,irr,red
31        num=num+1
32        denom=denom-1
33      som=som+1
34
35 a,b=2,6
36 rng,tot,gcd,Lirr,Lred=denombre(a,b)
37 if gcd==1:
38   print('{}/{} : fraction irreductible numero {}'.format(a,b,rng))
39 else:
40   print('{}/{} : fraction simplifiable numero {}'.format(a,b,tot-rng))
41 print('Nombre de fractions inferieures ou egales : {}'.format(tot))
42 print('Ratio irreductibles : {}%'.format(round(rng/tot*100,1)))
43 print('Fractions irreductibles par ordre croissant :')
44 for f in Lirr:
45   print('{}/{}'.format(f[0],f[1]),end=' ')
46 print('\nFractions simplifiables par ordre croissant :')
47 for f in Lred:
48   print('{}/{}'.format(f[0],f[1]),end=' ')
```

Il faut, pour ce programme, utiliser un module qui détermine le pgcd de deux entiers : `pgcd` en est une version récursive. La fonction `denombre` passe en revue les fractions concernées dans l'ordre du dénombrement et enregistre celles qui sont irréductibles dans une liste qu'il faut obtenir triée dans l'ordre numérique. Pour cette raison, j'enregistre les fractions sous la forme de listes de 3 éléments `[a,b,d]` où `d` est une valeur décimale (notation flottante des nombres réels) et j'insère la nouvelle fraction au bon endroit en me servant de `d` pour les ranger.
L'insertion proprement dite est confiée à la fonction `insere` qui détermine le rang de l'élément qui a une valeur décimale dépassant pour la première fois celle de l'élément à insérer. Remarquez que la liste `irr` peut être modifiée dans une autre partie que celle où elle a été déclarée sans qu'il soit nécessaire de renvoyer la dite liste par un bloc `return`. Ce n'était pas demandé mais le programme construit également la liste triée `red` des fractions réductibles et donne pour une telle fraction, son rang parmi les fractions réductibles.

```
deg                PYTHON
>>> from irreductibles import
2/6 : fraction simplifiable nu
Nombre de fractions inferieure
Ratio irreductibles : 78.3%
Fractions irreductibles par or
3 3/2 2/1 5/2 3/1 4/1 5/1 6/1
Fractions simplifiables par or
2/6 2/4 2/2 3/3 4/2
>>>
```