Devoir Maison d'ITC N°2
pour la semaine du 11/01/2024
à rendre en séance de TP

Consignes

- La qualité, la lisibilité et l'efficacité du code entreront pour une part importante dans l'appréciation de la copie. L'indentation du code doit figurer clairement sur la copie, il est fortement conseillé d'y ajouter une barre verticale sur la gauche afin de préciser clairement les portions de code ayant la même indentation.
- **Le crayon à papier ne sera pas corrigé.**
- **Il est rappelé également que recopier une correction sur internet est complètement inutile pour tout le monde.**

Problème Dans cet exercice, tout rationnel $\frac{a}{b}$ est représenté par la liste `[a, b]`.

On rappelle que pour tout rationnel $r \in \mathbb{Q}_+^*$, il existe $(a,b) \in \mathbb{N}^* \times \mathbb{N}^*$ tels que $r = \frac{a}{b}$ avec $\text{PGCD}(a,b) = 1$. Cette écriture s'appelle forme irréductible de r.

Partie I — Opérations sur les listes de rationnels

1. **1.1)** Écrire une fonction récursive `pgcd(a:int, b:int)->int` qui reçoit deux entiers `a` et `b` et qui renvoie leur pgcd.

   1.2) Pour $n \in \mathbb{N}^*$, on note :
   $$\phi(n) = \text{Card}\{k \in \mathbb{N} \mid 1 \leqslant k \leqslant n,\ \text{PGCD}(k,n) = 1\}.$$
   En utilisant la fonction précédente, écrire une fonction `ind(n:int)->int` qui reçoit un entier non nul n et qui renvoie $\phi(n)$.

2. **2.1)** Écrire une fonction `simpli(r:list)->list` qui reçoit un rationnel `r` et qui renvoie sa forme irréductible. *On veillera à ce que la liste retournée contienne, comme `L`, toujours des entiers.*

   2.2) En utilisant la fonction précédente, écrire une fonction *non récursive* `simplilist(L:list)->list` qui reçoit en argument une liste de rationnels et qui renvoie la liste de ces rationnels écrits sous forme irréductible.

   2.3) En utilisant la fonction précédente, écrire une fonction *récursive* `simplilistrec(L:list)->list` qui reçoit en argument une liste de rationnels et qui renvoie la liste de ces rationnels écrits sous forme irréductible.

3. **3.1)** Écrire une fonction *non récursive* `elimin(L:list, n:int)->list` qui reçoit en argument une liste de rationnels irréductibles, un entier et qui renvoie la liste `L` dans laquelle on a enlevé les rationnels dont le dénominateur est strictement supérieur à n.

   3.2) Écrire une fonction *récursive* `eliminrec(L:list, n:int)->list` qui reçoit en argument une liste de rationnels irréductibles, un entier et qui renvoie la liste `L` dans laquelle on a enlevé les rationnels dont le dénominateur est strictement supérieur à n.

Partie II — Les suites de Farey

Pour $n \in \mathbb{N}^*$, on appelle suite de Farey d'ordre n la suite croissante de rationnels écrits sous forme irréductibles appartenant à l'intervalle $[0,1]$ et dont le dénominateur est inférieur ou égal à n. On la note $\mathscr{F}_n$.On a par exemple :

$$\mathscr{F}_1 = \left\{\frac{0}{1}, \frac{1}{1}\right\};\ \mathscr{F}_2 = \left\{\frac{0}{1}, \frac{1}{2}, \frac{1}{1}\right\};\ \mathscr{F}_3 = \left\{\frac{0}{1}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{1}{1}\right\};\ \mathscr{F}_4 = \left\{\frac{0}{1}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{1}{1}\right\}.$$

Nous allons donner un procédé récursif de construction de $\mathscr{F}_n$ pour $n \in \mathbb{N}^*$. Pour $\frac{a}{b}$ et $\frac{c}{d}$ deux rationnels on appelle `médian` le rationnel défini par :

$$\text{Med}\left(\frac{a}{b}, \frac{c}{d}\right) = \frac{a+c}{b+d}.$$

Nous admettrons les résultats mathématiques suivants :

- Si r_1 et r_2 sont deux éléments consécutifs de $\mathscr{F}_n$ alors $\text{Med}(r_1, r_2)$ est un rationnel écrit sous forme irréductible.
- Pour $n \geqslant 2$, la suite $\mathscr{F}_n$ s'obtient en intercalant au milieu de chaque couple de termes consécutifs (r_1, r_2) leur médian si ce dernier a un dénominateur inférieur ou égal à n. Ainsi :

$$\mathscr{F}_1 = \left\{\frac{0}{1}, \frac{1}{1}\right\},\quad \mathscr{F}_2 = \left\{\frac{0}{1}, \frac{0+1}{1+1}, \frac{1}{1}\right\} = \left\{\frac{0}{1}, \frac{1}{2}, \frac{1}{1}\right\},\quad \mathscr{F}_3 = \left\{\frac{0}{1}, \frac{0+1}{1+2}, \frac{1}{2}, \frac{1+1}{2+1}, \frac{1}{1}\right\}.$$

4. Écrire une fonction `med(a:list, b:list)->list` qui reçoit deux rationnels et qui renvoie leur médian.
5. On considère le script :

```
def etape(L:list)->list:
    R = [L[0]]
    g = L[0]
    for d in L[1:]:
        m = med(g, d)
        R += [m, d]
        g = d
    return R
```

5.1) Dérouler la fonction `etape` sur la liste associée à $\mathscr{F}_3$. Comment s'obtient de manière générale la liste `etape(L)` à partir de `L` ?

5.2) Proposer une fonction récursive `etaperec(L:list)->list` qui réalise la même action que la fonction `etape(L)`.

6. On considère le script :

```
L = [[0, 1], [1, 1]]
for _ in range(3):
    L = etape(L)
```

6.1) Donner les différents contenus de `L`. Que faut-il faire pour obtenir $\mathscr{F}_4$?

6.2) À l'aide des questions précédentes, écrire une fonction `Farey(n:int)->list` qui renvoie la suite $\mathscr{F}_n$.

6.3) **i)** Écrire une fonction `Fareybis(n:int)` qui renvoie la suite $\mathscr{F}_n$ et qui à chaque étape filtre les éléments de la suite construite pour ne garder que les éléments qui appartiennent à une suite de FAREY.

ii) On désigne par ℓ_n la longueur de la suite $\mathscr{F}_n$. Montrer que :

$$\forall n \in \mathbb{N}^*, \quad \ell_{n+1} \leqslant a.\ell_n + b$$

où a et b sont deux entiers à préciser. En déduire une majoration de ℓ_n.

6.4) Comparer les deux fonctions qui donnent les suites de FAREY au niveau des parcours de listes et de l'espace mémoire utilisé.

Correction
du Devoir Maison d'ITC N°1

Solution

PARTIE I — OPÉRATIONS SUR LES LISTES DE RATIONNELS

1. 1.1)
```
def pgcd(a:int, b:int)->int:
    if b == 0:
        return a
    else:
        return pgcd(b, a%b)
```

1.2)
```
def ind(n:int)->int:
    N = 0
    for k in range(1, n+1):
        if pgcd(k, n) == 1:
            N += 1
    return N
```
```
>>> ind(10)
4
>>> ind(7)
6
```

2. 2.1)
```
def simpli(r:list)->list:
    a, b = r[0], r[1]
    d = pgcd(a, b)
    return [a//d, b//d]
```

2.2)
```
def simplilist(L:list)->list:
    Ls = []
    for r in L:
        Ls.append(simpli(r))
    return Ls
```
```
>>> simplilist([[2, 3], [3, 6]])
[[2, 3], [1, 2]]
```

2.3)
```
def simplilistrec(L:list)->list:
    if L == []:
        return L
    else:
        r = L[-1]
        return simplilistrec(L[:-1]) + [simpli(r)]
```
```
>>> simplilistrec([[2, 3], [3, 6]])
[[2, 3], [1, 2]]
```

La phase d'empilement correspond donc aux retraits successifs dans la liste `L`, et aux simplifications en fractions irréductibles. Lors du dépilement on ajoute fractions simplifiées dans `M`.

3. 3.1)
```
def elimin(L:list, n:int)->list:
    M = []
    for r in L:
        if r[1] <= n:
            M.append(r)
    return M
```
```
>>> elimin([[2, 3], [1, 1]], 2)
[[1, 1]]
```

3.2)
```
def eliminrec(L:list, n:int)->list:
    if len(L) == 0:
        return L
    else:
        r = L[-1]
        if r[1] <= n:
            return eliminrec(L[:-1], n) + [r]
        else:
            return eliminrec(L[:-1], n)
```
```
>>> eliminrec([[2, 3], [1, 1]], 2)
[[1, 1]]
```

PARTIE II — LES SUITES DE FAREY

4.
```
def med(a:list, b:list)->list:
    return [a[0]+b[0], a[1]+b[1]]
```
```
>>> med([0, 1], [1, 1])
[1, 2]
```

5. 5.1) On a $\mathscr{F}_3$= `[[0, 1], [1, 3], [1, 2], [2, 3], [1, 1]]`.

- On a au début $\mathscr{F}_3$= `[[0, 1], [1, 3], [1, 2], [2, 3], [1, 1]]`, `R = [[0, 1]]`, `g = [0, 1]`.
- À la première étape, `d = [1, 3]`, `m = med(g, d) = [1, 4]`. On ajoute ensuite ce rationnel à `R` ainsi que `d`, d'où `R = [[0, 1], [1, 4], [1,`

3]], g est alors changé en [[1, 3]].

- À la seconde étape, d = [1, 2], m = med(g, d) = [2, 5]. On ajoute ensuite ce rationnel à R ainsi que d, d'où R = [[0, 1], [1, 4], [1, 3], [2, 5], [1, 2]], g est alors changé en [[1, 2]].
- À la 3ème étape, d = [2, 3], m = med(g, d) = [3, 5]. On ajoute ensuite ce rationnel à R ainsi que d, d'où R = [[0, 1], [1, 4], [1, 3], [2, 5], [1, 2], [3, 5], [2, 3]], g est alors changé en [[2, 3]].
- À la dernière étape, d = [1, 1], m = med(g, d) = [3, 4]. On ajoute ensuite ce rationnel à R ainsi que d, d'où R = [[0, 1], [1, 4], [1, 3], [2, 5], [1, 2], [3, 5], [2, 3], [3, 4], [1, 1]], g est alors changé en [[1, 1]].

On peut vérifier avec le code le résultat obtenu :

```
>>> F3 = [[0, 1], [1, 3], [1, 2], [2, 3], [1, 1]]
>>> etape(F3)
[[0, 1], [1, 4], [1, 3], [2, 5], [1, 2], [3, 5], [2, 3], \
↪ [3, 4], [1, 1]]
```

De manière générale, la fonction etape retournera à partir d'une liste de rationnels, la même suite de rationnels où l'on aura intercalé les médians entre chaque terme pris deux à deux.

5.2)

```
def etaperec(L:list)->list:
    if len(L) < 2:
        return L
    else:
        a, b = L[-1], L[-2]
        return etaperec(L[:-1]) + [med(a, b)] + [a]
```

```
>>> etaperec(F3)
[[0, 1], [1, 4], [1, 3], [2, 5], [1, 2], [3, 5], [2, 3], \
↪ [3, 4], [1, 1]]
```

6. On considère le script :

```
from copy import deepcopy
L = [[0, 1], [1, 1]] # correspond à F1
for i in range(3):
    L = deepcopy(etape(L))
```

6.1)

```
>>> L = [[0, 1], [1, 1]]
>>> from copy import deepcopy
>>> for _ in range(3):
...     L = deepcopy(etape(L))
...     print(L)
...
[[0, 1], [1, 2], [1, 1]]
[[0, 1], [1, 3], [1, 2], [2, 3], [1, 1]]
[[0, 1], [1, 4], [1, 3], [2, 5], [1, 2], [3, 5], [2, 3], \
↪ [3, 4], [1, 1]]
```

Pour obtenir $\mathscr{F}_4$ il faut éliminer tous les rationnels dont le dénominateur est strictement supérieur à 4.

6.2)

```
def Farey(n:int)->list:
    L = [[0, 1], [1, 1]]
    for _ in range(n-1):
        L = deepcopy(etape(L))
    return elimin(L, n)
```

```
>>> Farey(4)
[[0, 1], [1, 4], [1, 3], [1, 2], [2, 3], [3, 4], [1, 1]]
```

6.3) **i)**

```
def Fareybis(n:int)->list:
    L = [[0, 1], [1, 1]]
    for i in range(2, n+1):
        L = deepcopy(elimin(etape(L), i)) # à l'étape \
↪ i on calcule Fi
    return L
```

```
>>> Fareybis(4)
[[0, 1], [1, 4], [1, 3], [1, 2], [2, 3], [3, 4], [1, \
↪ 1]]
```

ii) On remarque que si $\mathscr{F}_n$ contient ℓ_n éléments, pour fabriquer $\mathscr{F}_{n+1}$ on calcule $\ell_n - 1$ rationnels supplémentaires. Comme certains seront éliminés, on en déduit que $\ell_{n+1} \leqslant 2\ell_n - 1$. On remarque que $\ell_{n+1} - 1 \leqslant 2(\ell_n - 1)$ et comme $\ell_1 = 2$, on en déduit que $\ell_n - 1 \leqslant 2^{n-1}$. D'où $\boxed{\ell_n \leqslant 2^{n-1} + 1}$.

6.4) Si on filtre à chaque étape, on ne fait que les calculs nécéssaires. On utilise moins d'espace mémoire, mais cela nécessite plus de parcours de liste. Par contre si on ne filtre qu'à la dernière étape, la liste avant filtrage contient $\sum_{k=0}^{n-2} 2^k = 2^{n-1} - 1$ rationnels. On utilise moins de parcours, mais plus d'espace mémoire.