

# Premiers pas en OCaml

Florian Bourse

## 1 Nombres rationnels et suites récurrentes

Pour représenter un nombre rationnel  $\frac{a}{b}$ , nous allons utiliser un couple d'entiers  $(a, b)$  dont le premier élément est le numérateur et le second le dénominateur.

[0] Écrire une fonction `pgcd : int -> int -> int` qui prend en argument deux entiers  $a$  et  $b$  et calcule leur plus grand diviseur commun.

[1] Écrire une fonction `simplify : int * int -> int * int` qui prend en argument un nombre rationnel  $\frac{a}{b}$  et retourne la fraction irréductible qui lui est égal.

[2] Écrire les fonctions

`add_frac : int * int -> int * int -> int * int`

`mult_frac : int * int -> int * int -> int * int`

`div_frac : int * int -> int * int -> int * int`

qui prennent en argument deux nombres rationnels  $\frac{a}{b}$  et  $\frac{c}{d}$  et calculent respectivement leur somme, leur produit et leur quotient.

[3] On considère la suite  $x$  définie par ses deux premiers termes  $x_0 = 1$  et  $x_1 = 42$  et la relation de récurrence suivante :

$$x_{n+2} = \frac{x_{n+1} + 1}{x_n}$$

Calculer  $u_{13}$ ,  $u_{42}$ ,  $u_{1001}$ ,  $u_{2^{1000}-1}$ .

[4] On considère la suite  $u$  définie par son premier élément  $u_0$  et la relation de récurrence suivante :

$$u_{n+1} = \frac{u_n^2}{1 + 2u_n}$$

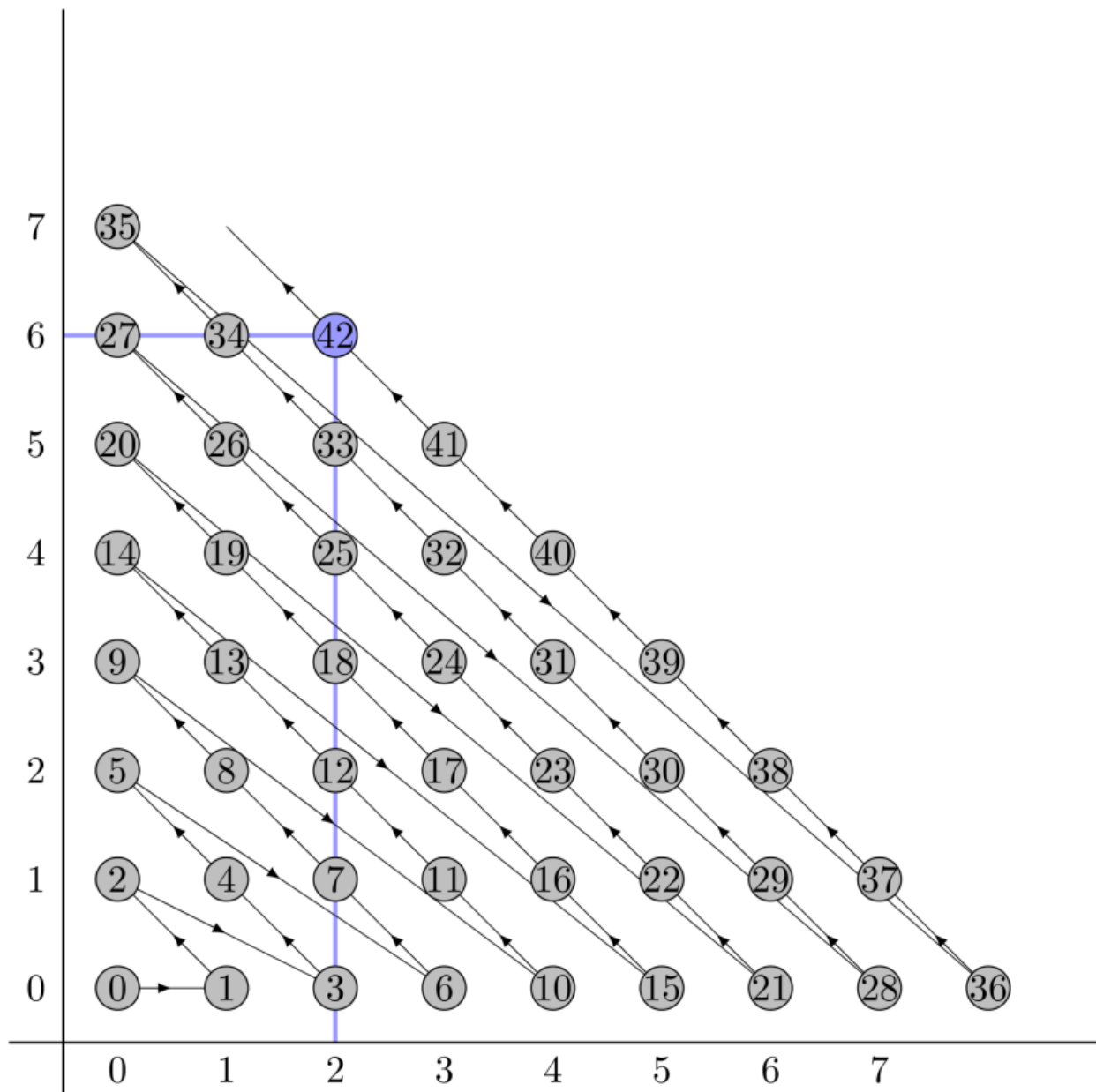
Écrire une fonction `u : int * int -> int -> int * int` qui prend en argument un nombre rationnel  $\frac{a}{b}$  et un entier  $n$  et qui renvoie  $u_n$  sachant que  $u_0 = \frac{a}{b}$ .

Tester cette fonction avec plusieurs valeurs et émettre une conjecture quand à la convergence de la suite  $u_n$ . Essayez de la prouver.

[5] Écrire une fonction `approx : int * int -> int * int` qui prend en argument un nombre rationnel  $\frac{a}{b}$  et qui renvoie le couple  $(n, d)$  vérifiant les propriétés suivantes :

- $n$  est l'arrondi à l'unité de  $\frac{a}{b}$  ;
- si  $\frac{a}{b}$  est un entier,  $d = 0$  ;
- sinon,  $d$  est le plus grand entier tel que

$$n - \frac{1}{d} \leq \frac{a}{b} \leq n + \frac{1}{d}$$



### 3 Décompositions en carré parfaits

[8] Écrire une fonction `decomp : int -> int` qui prend en entrée un entier  $n$  et qui renvoie le nombre de façons de décomposer  $n$  comme une somme de carrés parfaits distincts strictement positifs.

Exemple : `decomp 100` renvoie 3 car  $100 = 1^2 + 3^2 + 4^2 + 5^2 + 7^2 = 6^2 + 8^2 = 10^2$

`decomp 4` renvoie 1 car  $4 = 2^2$ , et  $4 = 1^2 + 1^2 + 1^2 + 1^2$  n'est pas compté car les termes ne sont pas distincts *Indication* : On pourra définir une fonction auxiliaire récursive

`decomp_aux : int -> int -> int`

qui prend en entrée un entier  $n$  et un entier  $i$  et qui résoud le problème en ne considérant pas comme termes possibles les carrés plus petits que  $i \times i$

Remarque : On ne cherche pas à trouver ces décompositions, juste à les dénombrer.

[9] Écrire une fonction `decomp_bis : int -> int` qui prend en entrée un entier  $n$  et qui renvoie le nombre de façons de décomposer  $n$  comme une somme de carrés parfaits strictement positifs. On autorise les doublons.

Exemple : `decomp_bis 4` renvoie 2 car cette fois, on compte  $4 = 1^2 + 1^2 + 1^2 + 1^2$

`decomp_bis 100` renvoie 1116.

### 4 Écriture décimale

Dans cette section, on pourra s'aider de la division euclidienne d'un nombre par 10. Si  $n = 10 \times d + u$ ,  $u$  est son chiffre des unités et  $d$  son nombre de dizaines. En OCaml,  $d$  est `n / 10` et  $u$  est `n mod 10`.

[10] Écrire une fonction `inverser : int -> int` qui prend en entrée un entier  $n$  et qui renvoie le nombre  $n'$  tel que les écritures décimales de  $n$  et  $n'$  sont inversées.

Exemple : `inverser 123` renvoie 321

*Indication* : On pourra créer une fonction récursive auxiliaire qui serait appelée sur les entrées suivantes par exemple :

|     |     |
|-----|-----|
| 123 | 0   |
| 12  | 3   |
| 1   | 32  |
| 0   | 321 |

[11] Écrire une fonction `palindrome : int -> bool` qui prend en entrée un entier  $n$  et qui renvoie `true` si et seulement si  $n$  est un palindrome, c'est-à-dire qui se lit de la même manière de gauche à droite et de droite à gauche.

[12] Écrire une fonction `solve : int -> int` qui prend en entrée un entier  $n$  et qui renvoie le plus petit  $k \geq n$  tel que  $k$  multiplié par son écriture inversée (en base 10) donne un palindrome.

Exemple : `solve 2018` renvoie 2021 car  $2021 \times 1202 = 2429242$