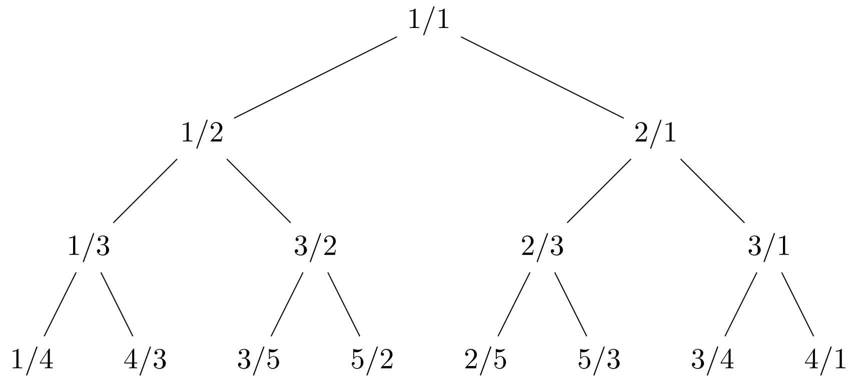
Problème 4 : Autour de l'énumération des fractions positives

21. La représentation de l'arbre de Calkin-Wilf jusqu'à la profondeur 3 est :



22. Pour pouvoir représenter des arbres de profondeur finie, on distingue les nuds des feuilles et on propose le type suivant :

```
type acw = F of fraction | N of acw * fraction * acw ;;
```

23. Montrons par récurrence sur le niveau d'exploration que si n/d est un nud de l'arbre alors $n \wedge d = 1$.
   - Au niveau 0, la seule fraction représentée est 1/1 et le résultat est immédiat.
   - On considère $k \in \mathbb{N}$ et on suppose que tous les nuds du niveau k sont de la forme n/d avec $n \wedge d = 1$.
     On peut donc considérer un couple de Bézout $(u, v) \in \mathbb{Z}^2$ associé au couple (n, d) :
     $$un + vd = 1$$
     On a alors :
     $$(u - v)n + v(n + d) = 1, \qquad u(n + d) + (v - u)d = 1$$
     Ces deux égalités assurent que $d \wedge n + d = n \wedge (n + d) = 1$. Les couples $(n + d, d)$, $(n, n + d)$ sont donc des couples de nombres entiers premiers entre eux. Ce qui permet de conclure.

24. On s'appuie sur l'algorithme d'Euclide :

```
let rec pgcd m n  =
  if m mod n = 0 then n else pgcd n (m mod n) ;;
```

25. On propose un code qui tient compte de la phrase "Au besoin, elle simplifie la fraction par $n \wedge d$". On pourrait faire la simplification dans tous les cas.

```
let fraction n d =
  if n <= 0 || d <= 0 then failwith "fracirreduc" else
   begin
    let delta = pgcd n d in match delta with
     | 1 -> {n=n;d=d}
     | _ -> {n=n/delta;d=d/delta}
   end;;
```

26. Avec les notations de l'énoncé, on pose :
$$N_1 = \frac{k}{l}, \quad N_2 = \frac{n}{d}$$
On suppose que les fils gauches sont égaux, alors :
$$\frac{N_1}{N_1 + 1} = \frac{N_2}{N_2 + 1}$$

Le résultat de la question 23 assure que les fractions $\frac{N_1}{N_1+1}$ et $\frac{N_2}{N_2+1}$ sont irréductibles.
Par unicité du représentant irréductible d'un nombre rationnel positif, on en déduit que $N_1 = N_2$.
Puisque les fractions $\frac{k}{l}$ et $\frac{n}{d}$ sont irréductibles et égales on en déduit que :

$$k = n, \quad l = d$$

On démontre de la même façon le résultat si les fils droits sont égaux.

27. Le résultat se démontre par récurrence sur $k \in \mathbb{N}$ en s'appuyant sur le fait que le fils gauche de N est $\frac{N}{1+N}$ (*Les détails ne sont pas donnés dans cette correction.*)
On montre de même que le nud provenant de d'une suite de k fils droits de N est $N + k$.

28. En s'appuyant sur l'arbre représenté dans la question 21, on obtient :

i	0	1	2	3	4	5	6	7
v_i	0	1	1/2	2	1/3	3/2	2/3	3/1

29.
- Pour $n + d = 2$, on a $n = d = 1$, puisque $v_1 = 1$ le résultat est vrai pour $n + d = 2$.
- On considère $k \in \mathbb{N}\backslash\{0, 1\}$ et on suppose que pour tout $(n, d) \in \mathbb{N}^{*2}$, tel que $n \wedge d = 1$ et $n+d \leq k$, la fraction $\dfrac{n}{d}$ apparaît dans l'arbre.
On considère alors deux nombres entiers naturels non-nuls n et d tels que $n \wedge d = 1$ et $n+d = k+1$.

— Si $\dfrac{n}{d} < 1$, alors n et $d - n$ sont deux nombres entiers naturels non-nuls premiers entre eux et :

$$n + (d - n) \leq k$$

l'hypothèse de récurrence assure alors que la fraction rationnelle $\frac{n}{d-n}$ apparaît dans l'arbre.
Or $\dfrac{n}{d}$ est le fils gauche de ce nud, la fraction $\dfrac{n}{d}$ apparaît donc dans l'arbre.

— Si $\dfrac{n}{d} > 1$, alors $n - d$ et d sont deux nombres entiers naturels non-nuls premiers entre eux et :

$$(n - d) + d \leq k$$

l'hypothèse de récurrence assure alors que la fraction rationnelle $\frac{n-d}{n}$ apparaît dans l'arbre.
Or $\dfrac{n}{d}$ est le fils droit de ce nud, la fraction $\dfrac{n}{d}$ apparaît donc dans l'arbre.

On peut donc conclure que pour tout couple de nombres entiers naturels non-nuls (n, d) premiers entre eux, la fraction n/d apparaît dans l'arbre.

30. On va montrer par récurrence que pour tout $k \in \mathbb{N}$, l'arbre de Calkin-Wilf de profondeur k, noté CW_k ne répète aucun nud.

- Pour $k = 0$, le résultat est immédiat.
- On considère $k \in \mathbb{N}$ et on suppose le résultat vrai pour CW_k.
Supposons que l'arbre CW_{k+1} admet deux nuds distincts de même valeur N.

— Si $N < 1$, il s'agit nécessairement de deux fils gauches. Or l'application :

$$^{+*}x\frac{x}{x+1}$$

est injective. Les deux nuds parents sont donc égaux et de profondeur strictement inférieure à k.
Cela contredit l'hypothèse de récurrence.

— Si $N > 1$, on conclut de la même façon en raisonnant sur les fils droit et en utilisant le caractère injectif de l'application :

$$^{+*}xx + 1$$

Pour tout $k \in \mathbb{N}$, l'arbre CW_k est donc sans répétition, ce qui permet de conclure.

31. Le résultat de la question 29 assure que l'application :

$$\mathbb{N}^{*+} i v_i$$

est surjective. Le résultat de la question 30 assure qu'elle est injective.
$v_0 = 0$ permet de conclure que la suite $v_{i i \in \mathbb{N}}$ définit une bijection de $\mathbb{N}$ dans $^+$.

32. On montre classiquement qu'à la profondeur k, il y a 2^k nuds d'indices variant de 2^k à $2^{k+1}-1$. Pour $i \in 2^k, 2^{k+1}-1$, les 2^k fils gauches sont les nuds d'indices pairs obtenus en multipliant l'indice du nud père par 2.
Puisque chaque fils droit est d'indice consécutif de l'indice du fils gauche, on obtient le résultat souhaité.

33. On a :

i	0	1	2	3	4	5	6	7	8	9
s_i	0	1	1	2	1	3	2	3	1	4

34. Le code suivant s'appuie sur la définition récursive de la fonction.

```
let rec stern i = match i with
 | 0                     -> 0
 | 1                     -> 1
 | n  when n mod 2 = 0 -> stern (n/2)
 | n                     -> let i = (n-1)/2 in (stern i) + (stern (i+1));;
```

35. Montrons par récurrence sur la profondeur k des nuds, que pour tout $i \in \mathbb{N}$:

$$v_i = \frac{s_i}{s_{i+1}}$$

- Pour $k = 0$, le résultat découle immédiatement de la définition de s_0 et s_1.
- On considère $k \in \mathbb{N}$ et on suppose le résultat vrai pour k.
Les nuds de profondeurs $k+1$ sont des nuds fils des nuds de profondeur k. En notant i l'indice d'un nud de profondeur k, il s'agit donc de montrer que :

$$v_{2i} = \frac{s_{2i}}{s_{2i+1}}, \quad v_{2i+1} = \frac{s_{2i+1}}{s_{2i+2}}$$

Or :

$$v_{2i} = \frac{v_i}{v_i + 1}$$

l'hypothèse de récurrence assure alors que :

$$v_{2i} = \frac{\frac{s_i}{s_{i+1}}}{\frac{s_i}{s_{i+1}} + 1} = \frac{s_i}{s_i + s_{i+1}} = \frac{s_{2i}}{s_{2i+1}}$$

De même :

$$v_{2i+1} = v_i + 1 = \frac{s_i}{s_{i+1}} + 1 = \frac{s_i + s_{i+1}}{s_i} = \frac{s_{2i+1}}{s_{2i}}$$

ce qui permet de conclure.

36. Dans le premier cas, on peut considérer $j \in \mathbb{N}^*$ tel que $i = 2j$ et $i+1 = 2j+1$.
Le résultat de la question 35 assure alors que :

$$v_i = \frac{s_{2j}}{s_{2j+1}}, \quad v_{i+1} = \frac{s_{2j+1}}{s_{2j+2}}$$

Soit encore :

$$v_i = \frac{s_j}{s_j + s_{j+1}}, \quad v_{i+1} = \frac{s_j + s_{j+1}}{s_{j+1}}$$

Or :

$$\frac{s_j}{s_j+s_{j+1}} = 1 - \underbrace{\frac{s_{j+1}}{s_j+s_{j+1}}}_{\in]0,1[}$$

Donc $\left\lfloor \frac{s_j}{s_j+s_{j+1}} \right\rfloor = 0$ et :

$$f\frac{s_j}{s_j+s_{j+1}} = \frac{1}{1+2\left\lfloor \frac{s_j}{s_j+s_{j+1}} \right\rfloor - \frac{s_j}{s_j+s_{j+1}}} = \frac{1}{1-\frac{s_j}{s_j+s_{j+1}}} = \frac{s_j+s_{j+1}}{s_{j+1}} = v_{i+1}$$

ce qui permet de conclure.

37. Le résultat de la question 27 permet d'établir par une récurrence rapide que la dernier nud à droite à la profondeur k, correspond à

$$v_{2^{k+1}-1} = k+1$$

On montre de même que le nud suivant, qui est le nud le plus à gauche de la profondeur $k+1$ vérifie :

$$v_{2^{k+1}} = \frac{1}{k+2}$$

Par ailleurs :

$$fk+1 = \frac{1}{1+2\lfloor k+1 \rfloor - k - 1} = \frac{1}{k+2}$$

ce qui permet de conclure.

38. Pour construire le chemin, on part de $\frac{n}{d}$ et on remonte jusqu'à la racine.

- Si $\frac{n}{d} < 1$ le parent du nud est $\frac{n}{d-n}$.
- Si $\frac{n}{d} > 1$ le parent du nud est $\frac{n-d}{d}$.

On réitère le procédé jusqu'à obtenir la racine.
On peut noter que l'on n'a pas besoin de la fonction `rev` pour obtenir le résultat souhaité.

```
let chemin frac =
 let rec cons res n d = match n,d with
  | 1,1                 -> res
  | n,d  when n/d<1 -> cons (G::res) n (d-n)
  | n,d                 -> cons (D::res) (n-d) d
 in cons [] frac.n frac.d;;
```

39. La fonction `direction` est construite sur le même principe :

```
let direction liste =
 let rec cons n d liste = match liste with
  | []       -> {n=n;d=d}
  | D::fin   -> cons (n+d) d fin
  | G::fin   -> cons n (n+d) fin
 in cons 1 1 liste;;
```

40. On obtient un ancêtre commun en atteignant le nud obtenu en parcourant le plus long chemin commun depuis la racine vers les deux nuds.
Le code de la fonction `ancetre` en découle.
On utilise la fonction auxiliaire `commun` qui détermine le plus long préfixe commun à deux listes.

```
let ancetre f1 f2 =
 let rec commun res l1 l2 = match l1,l2 with
  | [],l2                          -> res
  | l1,[]                          -> res
  | d1::fin1,d2::fin2 when d1 = d2 -> commun (d1::res) fin1 fin2
  | _,_                            -> res
 in let ch1,ch2 = chemin f1, chemin f2 in
 direction (commun [] ch1 ch2);;
```

41. Puisque v_i et v_{i+1} ont des indices consécutifs, si v_p est le premier ancêtre commun k' niveaux au-dessus des deux nuds, les chemins de v_p à v_i et de v_p à v_{i+1} sont respectivement de la forme :

$$[\mathtt{G}, \mathtt{D}, \cdots, \mathtt{D}], \quad [\mathtt{D}, \mathtt{G}, \cdots, \mathtt{G}]$$

Où les deux fins de listes sont de longueur $k' - 1$ et sont composées respectivement de `D` uniquement, de `G` uniquement.

42. Les résultats des questions 27 et 41 assurent que si $v_p = \dfrac{n}{d}$, alors :

$$v_i = \frac{n}{n+d} + k' - 1, \quad v_{i+1} = \frac{N}{1 + (k'-1)N} \text{ où } N = \frac{n+d}{d}$$

Puisque $\dfrac{n}{n+d} \in]0, 1[$, on a :

$$\lfloor v_i \rfloor = \left\lfloor \frac{n}{n+d} + k' - 1 \right\rfloor = k' - 1$$

Donc :

$$fv_i = \frac{1}{k' - \frac{n}{n+d}} = \frac{n+d}{k'(n+d) - n}$$

Par ailleurs :

$$\frac{N}{1 + (k'-1)N} = \frac{\frac{n+d}{d}}{1 + (k'-1)\frac{n+d}{d}} = \frac{n+d}{d + (k'-1)(n+d)} = \frac{n+d}{k'(n+d) - n}$$

ce qui permet de conclure.