

TD d'Éléments d'Algorithmique n° 2

Énumérations

Exercice 1. *Fonctions de $\{1, \dots, n\}$ dans $\{1, \dots, m\}$.*

Pour $n, m \in \mathbb{N}$ donnés, soit F_m^n l'ensemble de toutes les fonctions de $\{1, \dots, n\}$ dans $\{1, \dots, m\}$

1. Quelle est la cardinalité de F_m^n ?

Soit T_m^n l'ensemble des tableaux d'entiers de longueur n , dont les éléments appartiennent à $\{1, 2, \dots, m\}$.

Pour $f \in F_m^n$ donnée, soit $T_f \in T_m^n$ le tableau tel que pour tout $1 \leq i \leq n$, $T_f[i] = f(i)$.

2. Constaté que la fonction $f \mapsto T_f$ est une bijection entre F_m^n et T_m^n .

Les tableaux représentant ainsi les éléments de F_m^n peuvent être lus comme des nombres à n chiffres en base m , et ordonnées en tant que tels. Il s'agit de l'ordre *lexicographique* sur les séquences de n nombres compris entre 1 et m , le premier élément à partir de la gauche, c.à.d. l'élément d'indice 1 du tableau, étant le plus significatif. Le fait que 0 n'apparaisse dans ces séquences n'a pas d'importance. Dans la suite, \preceq_m^n désigne cet ordre.

3. Montrer que \preceq_m^n est un ordre total, et donner le plus petit et le plus grand élément de T_m^n relativement à \preceq_m^n .
4. Écrire un algorithme qui, étant donné un élément T de T_m^n différent du plus grand élément de T_m^n relativement à \preceq_m^n , renvoie son successeur relativement à \preceq_m^n , c.à.d. le plus petit élément de l'ensemble des majorants de T .
5. Écrire un algorithme de "génération exhaustive" de T_m^n , c.à.d. un programme qui renvoie (ou affiche) tous les éléments de T_m^n dans l'ordre \preceq_m^n .

★★ **Exercice 2.** *Tableaux d'inversions.*

Une permutation σ a une inversion à la place (i, j) si le couple (i, j) est tel que $i < j$ et $\sigma(i) > \sigma(j)$. On dit aussi que $(\sigma(i), \sigma(j))$ est une inversion de σ .

- Écrire à quelles places la permutation 3 2 4 1 a des inversions.

Le tableau d'inversions d'une permutation σ est le nombre d'inversions commençant à la place i , pour $i = 1, \dots, n$.

- Donner le tableau d'inversions de la permutation 6 3 2 1 5 4 7.
- Quelles sont les propriétés caractéristiques d'un tableau d'inversion à n éléments ? Donner une définition de tableau d'inversion.

Il y a une bijection entre les tableaux d'inversions et l'ensemble des permutations, c.à.d. à chaque tableau correspond un et un seul tableau d'inversions et viceversa. En effet à partir d'un tableau d'inversion on peut reconstruire la permutation en lisant le tableau de gauche à droite : l'entrée courante du tableau indique parmi les éléments encore non placés combien sont plus petits que l'élément recherché. Ainsi dans l'exemple du tableau 5,2,1,0,1,0,0, le 5 en première position indique que $\sigma(1)$ est plus grand que 5 éléments parmi $\{1, 2, 3, 4, 5, 6, 7\}$, c'est-à-dire que $\sigma(1) = 6$. Puis le 2 indique que $\sigma(2)$ est plus grand que 2 éléments parmi $\{1, 2, 3, 4, 5, 7\}$, c'est-à-dire que $\sigma(2) = 3$, etc...

- Écrire un algorithme qui prend un tableau d'inversion et qui reconstruit la permutation associée selon la bijection ci-dessus.
- Lister tous les tableaux d'inversion à 3 éléments dans l'ordre lexicographique et les permutations correspondantes par la bijection décrite ci-dessus.
- Écrire un algorithme qui engendre tous les tableaux d'inversion à n éléments dans l'ordre lexicographique.
- En utilisant la bijection ci-dessus traduire l'algorithme du point précédent sur les permutations à n éléments.

Exercice 3. Involutions.

Une permutation σ de $\{1, \dots, n\}$ est dite *involutive* (ou est appelée une *involution*) si pour tout $i \in \{1, \dots, n\}$, $\sigma(\sigma(i)) = i$. Elle est dite sans point fixe si pour tout $i \in \{1, \dots, n\}$, $\sigma(i) \neq i$. Une permutation σ de $\{1, \dots, n\}$ sera représentée par un tableau S tel que $S[i - 1] = \sigma(i)$ pour $i \in \{1, \dots, n\}$.

1. Les permutations suivantes de $\{1, \dots, 4\}$ sont-elles involutives ? Sont-elles sans point fixe ?

[1, 2, 3, 4]

[2, 1, 4, 3]

[4, 3, 2, 1]

[1, 3, 2, 4]

2. Écrivez un algorithme qui prend en entrée un entier $n > 0$ et un tableau S représentant une permutation σ de $\{1, \dots, n\}$, et qui retourne 1 si σ est involutive et 0 sinon.
3. Écrivez un algorithme qui prend en entrée un entier $n > 0$, et qui énumère toutes les involutions de $\{1, \dots, n\}$, dans l'ordre lexicographique.
- ★ 4. Combien y a-t-il d'involutions de $\{1, \dots, n\}$ sans point fixe ? On séparera le cas n pair du cas n impair.

Exercice 4. Permutations circulaires.

Une permutation σ de $\{1, \dots, n\}$ est dite *circulaire d'ordre k* (ou est appelée un *k-cycle*) s'il existe k entiers distincts a_1, \dots, a_k dans $\{1, \dots, n\}$, tels que $\sigma(a_i) = a_{i+1}$ pour $i \in \{1, \dots, k-1\}$ et $\sigma(a_k) = a_1$, et si pour tout $j \notin \{a_1, \dots, a_k\}$, $\sigma(j) = j$. Une permutation est dite *circulaire* (ou est appelée un *cycle*) s'il existe $k \in \{2, \dots, n\}$ telle qu'elle est circulaire d'ordre k . Une permutation σ de $\{1, \dots, n\}$ sera représentée par un tableau S tel que $S[i - 1] = \sigma(i)$ pour $i \in \{1, \dots, n\}$.

1. Énumérez à la main les 2-cycles de $\{1, 2, 3, 4\}$, dans l'ordre lexicographique. Faites de même pour les 3-cycles et les 4-cycles.
2. Écrivez un algorithme qui prend en entrée un entier $n > 0$, et qui énumère tous les 2-cycles de $\{1, \dots, n\}$ dans l'ordre lexicographique.
3. Écrivez un algorithme similaire pour les 3-cycles.
- ★ 4. Écrivez un algorithme qui prend en entrée un entier $n > 0$ et un entier $k > 0$, et qui énumère tous les k -cycles de $\{1, \dots, n\}$, dans l'ordre lexicographique.
- ★★ 5. Écrivez un algorithme qui prend en entrée un entier $n > 0$, un entier $k \in \{2, \dots, n\}$ et un tableau S représentant une permutation σ de $\{1, \dots, n\}$, et qui retourne 1 si σ est un k -cycle et 0 sinon.
6. En utilisant l'algorithme précédent, écrivez un algorithme qui prend en entrée un entier $n > 0$ et un tableau S représentant une permutation σ de $\{1, \dots, n\}$, et qui retourne 1 si σ est un cycle et 0 sinon. Cette question ne nécessite pas de savoir faire la question précédente.