TP n°4 - Permutations - Inversions d'un tableau

1 Suite du TP n° 3 : Permutations

On pourra récupérer le code correspondant aux questions 1 à 5 sur le site de la classe.

1.1 Énumération des permutations

1.1.1 Ordre sur les permutations

Q1 Écrire une fonction `trie_droite` qui prend en argument un tableau t et un entier k et qui trie en place la portion du tableau située à partir de l'indice k.

Q2 En déduire une fonction `suivant` qui prend en argument un tableau t représentant une permutation et modifie le tableau t en place de manière à obtenir la permutation suivante.

1.1.2 Affichage des permutations

Q3 Écrire une fonction `print_perm` prenant en argument une permutation et l'affiche sous la forme de votre choix.

Q4 En déduire une fonction `enumere` qui prend en argument un entier n et qui affiche toutes les permutations de $[\![0, n-1]\!]$. Combien y en a-t-il ?

1.2 Cycles

On appelle *cycle* une permutation σ tel qu'il existe k éléments distincts $x_0, \dots, x_{k-1}$ tels que le support de σ est $\{x_0, \dots, x_{k-1}\}$, et tels que $\sigma(x_0) = x_1$, $\sigma(x_1) = x_2$, $\dots$, $\sigma(x_{k-2}) = x_{k-1}$ et $\sigma(x_{k-1}) = x_0$.

Un tel cycle peut être représenté par la liste $[x_0; \dots; x_{k-1}]$.

On peut démontrer que toute permutation se décompose de manière unique comme un produit (commutatif) de cycles de supports disjoints. On représentera un produit de cycles sous la forme d'une liste de listes d'entiers.

Q5 Décomposer (à la main) la permutation `[|4; 6; 7; 1; 2; 5; 3; 0|]` en un produit de cycles disjoints.

Q6 Écrire une fonction `compose_cycle` prenant en argument un cycle c et un tableau t représentant une permutation pour laquelle on suppose que les éléments du cycle sont des points fixes, et qui modifie en place le tableau t de sorte que le tableau obtenu représente la composée du cycle et de la permutation initiale.

Q7 En déduire une fonction `produit_cycles` qui prend pour argument un produit de cycles et l'entier n et qui renvoie la permutation de $[\![0, n-1]\!]$ représentée par le produit de cycles.

Q8 On souhaite maintenant procéder à l'opération inverse. Écrire une fonction `cycles` qui prend en argument une permutation et qui décompose cette permutation en un produit de cycles de supports disjoints.

2 Inversion d'un tableau

Étant donnée un tableau d'entiers $t = [|t_0; t_1; \dots; t_{n-1}|]$, on appelle *inversion* de t tout couple $(i, j) \in [\![0, n-1]\!]$ tel que $i < j$ et $t_i > t_j$.

Q1 Écrire une fonction `nb_inversions` qui calcule le nombre d'inversions en implémentant un algorithme naïf. Quelle est sa complexité ?

Q2 Réecrire cette fonction en adoptant une stratégie « diviser pour régner ». On pourra pour cela adapter l'algorithme de tri fusion ; on s'autorisera l'utilisation de tableaux supplémentaires, tout en veillant à garder une complexité raisonnable.