Enumération du groupe symétrique

On représente une permutation σ dans $\mathfrak{S}_n$ par la liste $[\sigma(1), \sigma(2), ..\sigma(n)]$.

1. Ecrire un programme qui prend en entrée la liste $[\sigma(1), \sigma(2), ..\sigma(n)]$ et fournit en sortie la décomposition en cycles disjoints de σ.
2. De même avec la signature de σ.

On munit $\mathfrak{S}_n$ de l'ordre lexicographique :
Soient σ une permutation correspondant à la liste $[i_1, ..., i_n]$ et τ une permutation correspondant à la liste $[j_1, ..., j_n]$; $\tau < \sigma$ si il existe $k < n$ tel que $i_1 = j_1$, ..., $i_k = j_k$ et $j_{k+1} < i_{k+1}$. On remarquera qu'il s'agit d' un ordre total.

On range les permutations de $\mathfrak{S}_n$ selon l'ordre lexicographique : Par exemple, pour $n = 3$:
$[1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2], [3, 2, 1]$.

3. Etablir un algorithme qui prend en entrée n et sort la liste des permutations de $\mathfrak{S}_n$ rangées dans l'ordre lexicographique. Le faire tourner et évaluer jusqu'à quel entier n il cesse de tourner en quelques minutes. Pour ces valeurs, faire un graphique représentant le temps nécessaire à cette énumération.
4. Soient σ une permutation correspondant à la liste $[i_1, ..., i_n]$ et τ une autre permutation correspondant à la liste $[j_1, ..., j_n]$. Ecrire un programme `minlex` qui prend en entrée les deux listes représentant τ et σ et en sortie le plus petit des deux.
5. En déduire un algorithme pour calculer le numéro de σ dans la liste. Le programmer (`numlex`). Le tester en comparant avec les cas des petites valeurs de n où vous avez énuméré la liste .
6. On suppose maintenant que τ est le successeur de σ dans la liste. Démontrer qu'il existe $k < n$ tel que $\forall l < k$, $i_l = j_l$, et $i_k < j_k$. Montrer alors que :
   – $\{i_k, ..., i_n\} = \{j_k, ..., j_n\}$, on note α cet ensemble ordonné par $<$.
   – La liste $[j_{k+1}, ..., j_n]$ est le plus petit élément de de $\mathfrak{S}_{\alpha \setminus \{j_k\}}$ et $[i_{k+1}, ..., i_n]$ est le plus grand de $\mathfrak{S}_{\alpha \setminus \{i_k\}}$.
   – En déduire que $j_{k+1} < ... < j_n$ et que $i_{k+1} > .... > i_n$.
   – Démontrer que k est le plus grand entier l telque $i_l < i_{l+1}$.
   – Etudier la réciproque.
   – Ecrire un algorithme (`succlex`) qui calcule le successeur de σ dans la liste. Le tester en utilisant les programmes précédents.
7. Ecrire un algorithme (`predlex`) qui calcule le prédecesseur de σ dans la liste. Le tester en utilisant les programmes précédents.

Points fixes des permutations

Pour chaque permutation σ dans $\mathfrak{S}_n$, on note f_σ le nombre de points fixes de σ, c'est-à-dire le cardinal de l'ensemble $\{i; i \in \{1, ..., n\} \ / \ \sigma(i) = i\}$. On tire une permutation σ au hasard de façon uniforme dans $\mathfrak{S}_n$. On considère la variable aléatoire U_n, nombre de points fixes de la permutation tirée (si on tire la permutation σ, U_n prend la valeur f_σ).

8. Etablir un programme `Fix` qui prend en entrée une permutation sous forme de tableau et calcule le cardinal de son ensemble de points fixes.

9. Calculer le nombre moyen du nombre de points fixes des permutations de $\mathfrak{S}_n$ pour de petites valeurs de n.

10. Quel est ce nombre en fonction de n ? Le démontrer.

Nombre d'inversions des permutations

Pour chaque permutation σ dans $\mathfrak{S}_n$, on appelle *inversion* de σ tout couple (i, j) tel que $1 \leq i < j \leq n$ et $\sigma(j) < \sigma(i)$.

11. Démontrer que le nombre d'inversions d'une permutation est au plus $\begin{pmatrix} n \\ 2 \end{pmatrix}$. Exhiber une permutation dont le nombre d'inversions est 0 et une dont le nombre d'inversions est $\begin{pmatrix} n \\ 2 \end{pmatrix}$.

12. Programmer une fonction qui prend en entrée une permutation σ dans $\mathfrak{S}_n$ représentée par par la liste $[\sigma(1), \sigma(2), ..\sigma(n)]$ et retourne son nombre d'inversions.

13. Programmer le nombre moyen de points fixes des permutations de $\mathfrak{S}_n$ pour de petites valeurs de n.

Soit $u = [u_1, ..., u_n]$ une liste ne contenant que les chiffres 1,2,...,n pris chacun exactement une fois. On lui associe un vecteur $\omega_u = (\omega_1, ..., \omega_{n-1})$ défini par, pour tout i entre 1 et $n-1$, ω_i est le nombre de $j > i$ situés à gauche de i dans u. Ce vecteur est appelé *vecteur des inversions* de u. Par exemple pour $n = 5$, le vecteur des inversions du mot 31452 est $(1, 3, 0, 0)$.

14. Programmer une fonction qui prend en entrée une permutation σ dans $\mathfrak{S}_n$ représentée par par la liste $[\sigma(1), \sigma(2), ..\sigma(n)]$ et retourne son vecteur d'inversions.

15. Etablir un programme qui prend en entrée un entier naturel n et retourne la liste des vecteurs d'inversions des permutations de $\mathfrak{S}_n$ rangées selon l'ordre lexicographique. Le faire tourner pour des petites valeurs de n. Que constatez-vous?

16. Soit $\omega = (\omega_1, ..., \omega_{n-1})$ un vecteur d'entiers tel que, pour tout i, on ait l'encadrement $0 \leq \omega_i \leq n - i$. Démontrer que le programme ci dessous construit une permutation σ dans $\mathfrak{S}_n$ (représentée par par la liste $[\sigma(1), \sigma(2), ..\sigma(n)]$) dont le vecteur d'inversion est ω.

    $L := [n]$

    `Pour` i `de` $(n-1)$ `à` 1 `insérer` i `dans` L `à la` $\omega_i + 1$ `place.`

    En déduire que l'application qui envoie une permutation sur son vecteur d'inversions est injective.

17. Exprimer le nombre d'inversions de la permutation σ en fonction des coordonnées du vecteur ω_{u_σ}.

18. Soit k un nombre compris entre 0 et $\begin{pmatrix} n \\ 2 \end{pmatrix}$. Démontrer qu'il existe une permutation σ dont le nombre d'inversions est égal à k.

19. Déterminer exactement le nombre moyen du nombre d'inversions des permutations de $\mathfrak{S}_n$.
(Pour σ une permutation représentée par sa liste $[\sigma(1), \sigma(2), ..\sigma(n)]$, on pourra considérer $\tilde{\sigma}$ la permutation représentée par la liste $[\sigma(n), \sigma(n-1), ..\sigma(1)]$).

20. On définit une relation $\preceq$ sur $\mathfrak{S}_n$ en posant : $\sigma \preceq \tau$ si toute décroissance de σ est une décroissance de τ.

(i). Démontrer que $\preceq$ est une relation d'ordre.

(ii). Dessiner le diagramme de Hasse de la relation $\preceq$ sur $\mathfrak{S}_3$.

(iii). Soit E une partie de $\{(i,j) \ ; \ (i,j) \in \{1,...,n\}^2 \ / \ i < j\}$. Démontrer qu'il existe une permutation α dans $\mathfrak{S}_n$ d'ensemble de décroissances E si et seulement si il vérifie pour tous (i,j,k) tels que : $1 \leq i < j < k \leq n-1$

P1 : $(i,j) \in E \wedge (j,k) \in E, \Rightarrow (i,k) \in E$,

P2 : $(i,k) \in E \Rightarrow (i,j) \in E \vee (j,k) \in E$.

(iv). Démontrer qu'on obtient ainsi un treillis .
On écrira un algorithme qui, à partir des représentations de deux permutations sous forme de listes calcule le mot représentant la borne supérieure de ces deux mots pour l'ordre $\preceq$.
Faire de même pour la borne inférieure.