

Langages de programmation de haut niveau

Travaux Pratiques

Année 2019/2020

- Permutations

1 - Les deux représentations

1.a - Rotation d'un cran

Écrire, de la façon la plus concise possible, le code d'une fonction effectuant une rotation vers la droite d'un tableau (appliqué à [12,5,7,21,6] elle doit fournir [6,12,5,7,21]), puis une autre effectuant une rotation vers la gauche.

1.b - Permutations

Une permutation (et en particulier une rotation) peut être représentée soit par une liste des nouveaux indices soit par une matrice carrée ayant exactement un 1 par ligne et par colonne, le reste étant des zéros.

Par exemple `c(2,4,5,1,3)` représente la permutation qui à 1 associe 2, à 2 associe 4, à 3 associe 5,... Sa représentation matricielle est visible ci-contre

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	0	1	0	0	0
[2,]	0	0	0	1	0
[3,]	0	0	0	0	1
[4,]	1	0	0	0	0
[5,]	0	0	1	0	0

Une telle matrice permet d'effectuer la permutation correspondante d'un vecteur de nombres grâce à une multiplication de matrices. Écrire le code des deux fonctions permettant de transformer chaque représentation en l'autre. Donner des exemples d'utilisation.

1.c - Applications

Écrire une fonction qui applique une permutation sur un vecteur de valeurs numériques (entières ou réelles) pour chacune des deux représentations.

Écrire une fonction permettant de composer deux permutations (pour chacune des deux représentations).

Écrire de même deux fonctions donnant la permutation inverse c'est à dire la bijection réciproque.

1.d - Ordre

L'ordre d'une permutation est le plus petit entier p telle que composer p fois cette permutation donne l'identité.

Proposer différentes programmations d'une fonction qui détermine l'ordre d'une permutation : en utilisant la notation matricielle ou la notation vecteur, en utilisant des boucles explicites ou avec une écriture plus concise.

2 - Étude

2.a - génération

Écrire une fonction prenant en paramètre une taille n et un nombre de tirages p et produisant un vecteur de p permutations des entiers de 1 à n choisies aléatoirement.

2.b - graphiques

Comparer graphiquement le temps d'exécution de vos différentes fonctions de calcul d'ordre lorsque la taille de la permutation augmente (ne pas hésiter à tester de grandes tailles). Les temps comparés seront des moyennes sur un échantillon aléatoire (discuter du choix de p)

3 - Rendu

Déposer sur madoc un court document comportant le code de toutes les fonctions (exercices 1 et 2) avec quelques exemples d'application et les graphiques obtenus en 2.b, assortis d'un rapide commentaire. Le travail est à faire en binôme de préférence.