

INITIATION À LA PROGRAMMATION EN C (L1 CPEI)*

TP 7 : PERMUTATIONS

13/03/2019



À noter : Dans ce TP, nous notons par \mathfrak{S}_N l'ensemble des permutations de l'ensemble $\{1, \dots, N\}$, où $N \geq 1$.

§1. ÉCHAUFFEMENT

Exercice 1. (1) Listez toutes les permutations de \mathfrak{S}_7 , \mathfrak{S}_8 , et \mathfrak{S}_9 . Combien y en a-t-il ?
(2) Comptez à présent le nombre de permutations de \mathfrak{S}_7 , \mathfrak{S}_8 , et \mathfrak{S}_9 , sans point fixe. On dit que x est un *point fixe* d'une fonction f si $f(x) = x$.

Exercice 2. On dit qu'un générateur aléatoire de nombres est *uniforme* si tous les résultats ont la même probabilité. Par exemple, un dé uniforme ne renvoie aucun nombre plus souvent que les autres. Les deux questions qui suivent sont indépendantes.

- (1) Considérez que vous disposez de la fonction `int flip()` qui renvoie soit 0 soit 1, de façon uniforme. En utilisant seulement la fonction `flip()`, écrivez une fonction qui renvoie aléatoirement 0, 1 ou 2, de façon uniforme.
- (2) Écrivez une fonction `void shuffle(int a[], int len)` qui permute uniformément les éléments d'un tableau.

*Cours donné par prof. Roberto Amadio. Moniteur 2019 : Cédric Ho Thanh. TPs/TDs basés sur ceux des précédents moniteurs : Florian Bourse (2017), Antoine Dallon (2018). Autres contributeurs : Juliusz Chroboczek, Gabriel Radanne.

§2. UNE PETITE ÉNIGME

L'ignoble professeur C. Lang a capturé 100 étudiants, et promet de les libérer à condition qu'ils réussissent une épreuve.

Il numérote les étudiants de 1 à 100, et va les faire passer dans une salle qui contient 100 boîtes, également numérotées de 1 à 100. Chacune de ces boîtes contient un numéro, qui a été attribué suivant une permutation aléatoire $\sigma \in \mathfrak{S}_{100}$. Ainsi, la boîte n contient le numéro $\sigma(n)$. Cependant, cette permutation n'est connue que du prof. Lang...

A tour de role, les étudiants entrent dans cette salle, et doivent essayer de retrouver leur numéro en ouvrant au plus 50 boîtes. Si un seul des étudiants échoue ils resteront tous enfermés.

La situation semble désespérée, cependant il existe une stratégie qui maximise les chances de survie des étudiants. Les exercices de cette section devraient permettre de les aider.

Exercice 3. Si chaque étudiant procède au hasard, quelle est la probabilité qu'ils s'en sortent tous ?

On dit qu'une permutation $\tau \in \mathfrak{S}_N$ contient un *cycle de taille c* (où $c \geq 1$) s'il existe un nombre $1 \leq i \leq N$ tel que $\tau^c(i) = i$, mais tel que pour tout $1 \leq k < c$, $\tau^k(i) \neq i$.

Exercice 4. (1) Soit $\tau \in \mathfrak{S}_N$ une permutation. Montrez que tout nombre $1 \leq i \leq N$ fait partie d'un cycle, c'est-à-dire qu'il existe un $c \geq 1$ tel que $\tau^c(i) = i$.
(2) Pour des petits entiers N , comptez le nombre de permutation sur N éléments qui n'ont pas de cycle de taille strictement plus grande que $N/2$.

On propose la stratégie suivante : chaque mathématicien va commencer par ouvrir la boîte qui porte son numéro. Ensuite, il va ouvrir la boîte qui porte le numéro inscrit à l'intérieur de cette première boîte, et ainsi de suite jusqu'à trouver son propre numéro, ou à avoir ouvert 50 boîtes.

Exercice 5. (1) Si tous les étudiants appliquent cette stratégie, alors il vont s'en sortir si et seulement si il n'y a pas de cycle de taille plus grande que $50 = 100/2$ dans σ (la permutation secrète du prof. Lang). Pourquoi ?
(2) Au fait, combien y a-t-il de permutation dans \mathfrak{S}_{100} ?
(3) Les énumérer toutes serait beaucoup trop long, même pour un ordinateur. On va donc procéder à une simulation. On tire aléatoirement une permutation dans \mathfrak{S}_{100} , et on regarde si elle contient un cycle de taille 51 ou plus. Si oui, alors nos étudiants vont malheureusement y rester... sinon, ils sont sauvés ! En répétant ce procédé un nombre suffisant de fois, en en comptant le nombre de fois où l'issue est favorable, on obtient une approximation des chances de survie des étudiants. Avec ces instructions, écrivez un programme qui calcule une telle approximation.