# Roues de la roulette

laurent.jospin.59@free.fr, `http://jospin.lstl.fr`

Lycée Saint-Louis, Paris

On se donne $n$ valeurs $(x_i)_{i=0..n-1}$ quelconques et on associe à chacune de ces valeurs une valeur numérique $(w_i)_{i=0..n-1}$. On dira qu'une valeur $x_i$ a le poids $w_i$. On note $W$ la somme des poids $(w_i)$. On cherche à écrire un algorithme qui choisit aléatoirement une valeur parmi $(x_i)$ avec des probabilités $(p_i)$ proportionnelles aux poids $(w_i)$.

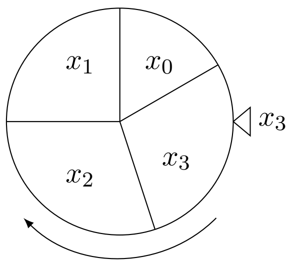


FIGURE 1 – Illustration de la roulette pour quatre valeurs placées sur quatre secteurs dont les angles au centre sont proportionnels aux poids. Après rotation, cette roue s'est arrêtée sur la valeur $x_3$

On utilisera dans cette activité la fonction `rand` de `stdlib`, sans arguments qui renvoie un entier entre 0 et `RAND_MAX`. On pourra initialiser le générateur (sa graine) en fonction du temps à l'aide de `srand( time( NULL ) );` (qui ne renvoie rien) dans la fonction `main` en incluant le fichier `time.h`.

Les nombres $p_i$ et $w_i$ seront stockés dans des tableaux (structure indicée) initialisés de la façon suivante **`double`** `poids[] = {`$w_0$`,`$w_1$`, ... ,`$w_{n-1}$`}`, de même pour les valeurs dans `valeurs`. On pourra accéder à la ième valeur d'un tableau grâce à `poids[i]` par exemple. Pour un tableau d'entier, le type d'un argument est **`int`**`*` ou **`int`** `...[]` et pour un tableau de flottant **`double`**`*` ou **`double`** `...[]`. Il n'y a pas de moyen simple de connaître la taille d'un tableau en C, elle doit donc être passée également en argument ou définie en globale si elle est caractéristique du problème. Elle sera ici définie en argument chaque fois que nécessaire par un argument nommé `taille` et faisant référence à $n$ dans le texte.

## 1 Avec les probabilités cumulées

**1.** Justifier que $\forall i \in [\![0; n-1]\!], p_i = \frac{w_i}{W}$.

**2.** Ecrire une fonction qui prend en argument un tableau de nombres à virgule flottante et sa taille et en renvoie la somme. Prouver la correction de la fonction écrite.

**3.** Ecrire une fonction `indice_choisi` qui prend en argument un tableau contenant les poids, la taille du tableau et une valeur `proportion` comprise dans $[0, 1[$ et renvoie le plus petit indice $i$ telle que $p \leqslant \frac{\sum_{j \leqslant i} w_j}{\sum w_j}$ où $p$ est la valeur d'argument `proportion`.

**4.** Ecrire une fonction basée sur `rand()` et `RAND_MAX` qui renvoie un nombre à virgule aléatoire de l'intervalle $[0; 1[$ selon un tirage uniforme.

**5.** Montrer que si on tire aléatoirement un réel $p \in [0,1[$ et qu'on prend la valeur correspondante dans le tableau `valeurs` de l'indice renvoyé par la fonction précédente alors on obtient bien une valeur aléatoire de `valeurs` avec des probabilités proportionnelles aux poids.

Ecrire la fonction correspondante nommée `choix_pondere_sondage_lineaire` qui prendra en argument `poids`, `valeurs` et `taille`.

**6.** Etudier la complexité de la fonction précédente. Quelle amélioration pourrait-on faire lorsqu'on tire un grand nombre de valeurs avec les mêmes tableaux `poids` et `valeurs` ? Cela améliorerait-il la complexité asymptotique dans le pire cas de la fonction ?

# 2 Avec un algorithme stochastique

**7.** Ecrire une fonction qui détermine le maximum d'un tableau de valeurs flottantes passé en argument (ainsi que sa taille). Prouver la correction de la fonction écrite.

**8.** Soit $M = \max\left(w_k\right)_{k=0..n-1}$. On considère l'algorithme suivant : on tire équiprobablement un entier $i$ de l'intervalle d'entiers $\llbracket 0, n-1 \rrbracket$. On tire ensuite uniformément un réel $p \in [0,1[$ et si $p < \frac{w_i}{M}$, on renvoie la valeur d'indice $i$ sinon on recommence les deux tirages jusqu'à ce que cette condition soit vérifiée.

Ecrire une fonction `choix_pondere_stochastique` qui prend en argument le tableau des valeurs, celui des poids, la taille et le maximum des poids et qui réalise l'algorithme décrit.

**9.** Réaliser le graphe de flot de contrôle de l'algorithme précédent. Peut-on prouver la terminaison de la fonction précédente ?

**10.** Ecrire une fonction `test` prenant en argument les poids, les valeurs, la taille et un nombre de répétitions et qui affiche les fréquences d'apparition avec les deux fonctions `choix_pondere` ainsi que les probabilités.

*Pour les deux dernières questions, il n'est pas attendu une rigueur parfaite.*

**11.** Déterminer la probabilité qu'un tour quelconque renvoie la valeur $x_i$. En déduire que l'algorithme choisit aléatoirement une valeur parmi $(x_i)$ avec des probabilités $(p_i)$ qui sont bien, *in fine*, proportionnelles aux poids $(w_i)$.

**12.** Calculer la probabilité qu'un tour boucle ne permette pas de renvoyer une valeur. Déterminer la complexité de l'algorithme.