Informatique tronc commun
Devoir n° 02

À rendre le 05 et le 06 mars 2017

Ce devoir est à réaliser *individuellement* pendant les vacances d'hiver.

Vous écrirez les fonctions demandées dans le langage Python (version 3) et rendrez une version informatique par mail de votre travail, avant le 05 mars à 22h00. Le 06 mars vous rendrez sur papier la réponse à la question 4.

Vos programmes seront en grande partie testés automatiquement, il est donc impératif que vous respectiez scrupuleusement les noms des fonctions précisés dans l'énoncé.

Enfin, le script que vous rendrez s'appellera `d02m-nom.py`, où `nom` est à remplacer par votre nom (sans majuscule, espace, accent ou caractère spécial). De même, les figures demandées aux questions 8 et 9 s'appelleront `d02m-nom-marche.png` et `d02m-nom-excursions.png`, respectivement.

I. Lecture d'un fichier de nombres.

On suppose que l'on dispose d'un fichier `nombres.txt` contenant des entiers naturels séparés pas des blancs (on rappelle qu'un blanc est soit un espace, soit une tabulation `\t` soit un retour à la ligne `\n`).

Pour faciliter votre travail, un exemple de tel fichier est disponible sur le site de classe. Nous vous encourageons fortement à tester vos fonctions sur des exemples dont vous aurez calculé le résultat à la main.

Q1 Écrire une fonction `somme()`, sans argument, renvoyant la somme de tous les entiers contenus dans ce fichier.

Q2 Pour chaque ligne du fichier, on effectue le produit des entiers de cette ligne. Écrire une fonction `moyenne()`, sans argument, renvoyant la moyenne de tous ces produits.

Q3 Si i est un entier inférieur au nombre de lignes du fichier, on note s_i la somme des entiers de la ligne i. Écrire une fonction `inversion()`, sans argument, retournant le nombre de couples (i, j) tels que $i < j$, et $s_j < s_i$.

Les invariants de boucle seront impérativement précisés en commentaires, dans le script renvoyé.

Q4 Dans cette dernière question, on suppose que le fichier contient n lignes, et que chaque ligne ne contient qu'un entier. Quelle est en fonction de n la complexité de l'algorithme `inversion()` ?

On supposera que les opérations de lecture dans le fichier se font en temps constant. Ainsi, l'opération consistant à lire tous les entiers du fichier et à les stocker un par un dans un tableau sera supposée avoir une complexité en $O(n)$.

Pour plus de clarté, vous recopierez le code de la fonction `inversion()` avant d'étudier sa complexité.

II. Excursions hors de zéro d'une marche aléatoire sur $\mathbb{Z}$.

On dispose en `Python` de la fonction `randrange` dans la bibliothèque `random`. Elle prend deux arguments entiers a et b, et renvoie un entier tiré uniformément dans $[\![a;b[\![$. On peut considérer que deux appels successifs de cette fonction donnent des tirages indépendants l'un de l'autre.

Une variable aléatoire R suit la loi de *Rademacher* si R ne peut prendre que 1 ou -1 comme valeurs, avec $P(R=1)=P(R=-1)=\dfrac{1}{2}$.

Étant donné une suite $(R_n)_{n\in\mathbb{N}}$ de variables aléatoires indépendantes et suivant chacune la loi de Rademacher, on considère la suite $(S_n)_{n\in\mathbb{N}}$ définie par

$$S_0=0 \quad \text{et} \quad \forall n\in\mathbb{N}^*,\ S_n=\sum_{k=1}^{n} R_n.$$

Notamment, si $n\in\mathbb{N}$, $S_{n+1}=S_n+R_{n+1}$. On peut voir cette suite $(S_n)_{n\in\mathbb{N}}$ comme une *marche aléatoire* sur $\mathbb{Z}$, et représenter son évolution en traçant les points (k,S_k) pour k dans une plage raisonnable.

Étant donné une telle suite $(S_n)_{n\in\mathbb{N}}$ et deux entiers naturels a,b avec $a<b$, on dit que la marche aléatoire S réalise une excursion hors de zéro entre a et b si

$$S_a=S_b=0 \quad \text{et} \quad \forall k\in[\![a+1;b[\![,\ S_k\neq 0.$$

Vous trouverez sur le site de la classe :

- un fichier `marche.txt` contenant un exemple de telle marche ($n=500$, les valeurs sont séparées par des espaces) ;
- un fichier `excursions.txt` contenant les excursions pour cette marche (une excursion par ligne) ;
- un fichier `d02m-nom-marche.png` représentant cette marche (vous devrez renvoyer une figure similaire) ;
- un fichier `d02m-nom-excursions.png` représentant les excursions hors de zéro de cette marche (vous devrez renvoyer une figure similaire).

Q5 Écrire une fonction `R()`, sans argument, renvoyant une réalisation d'une variable aléatoire de loi de Rademacher.

Q6 Écrire une fonction `marche(n)`, prenant en argument un entier naturel n, et renvoyant une liste contenant une réalisation de $[S_0,\dots,S_n]$.

Q7 Écrire une fonction `excursions(m)`, prenant en argument une liste d'entiers `m`, et renvoyant la liste des excursions pour la marche `m`. Par exemple, si

$$\texttt{m}=[0,1,0,-1,-2,-1,-2,-1,0,1,2],$$

la fonction `excursions(m)` renverra la liste

$$\texttt{les\_e} = [\ [0,1,0]\ ,\ [0,-1,-2,-1,-2,-1,0]\ ].$$

Q8 Écrire une fonction `trace_marche(m,nom_de_fichier)` ne renvoyant rien et enregistrant dans le fichier `nom_de_fichier` le tracé de la marche aléatoire dont les valeurs sont données dans `m`.

Vous enverrez à votre enseignant un tracé produit par cette fonction pour $n = 500$.

Q9 Écrire une fonction `trace_excursions(les_e,nom_de_fichier)` ne renvoyant rien et enregistrant dans le fichier `nom_de_fichier` le tracé des excursions données dans `les_e`.

Vous enverrez à votre enseignant un tracé produit par cette fonction, pour la même marche que la question précédente.

Remarque : on n'hésitera pas à retirer quelques réalisations de la marche de manière à avoir un tracé lisible et un nombre d'excursions raisonnable.