Lycée Louis-Le-Grand, Paris 2015/2016
MPSI 4 – Informatique commune
G. Dewaele, A. Troesch

TP n° 7 : Algorithmes élémentaires classiques

Nous étudions dans ce TP plusieurs algorithmes indépendants, qui se retouvent, sous cette forme ou de façon adaptée, à la base de nombreux algorithmes plus complexes.

Exercice 1 – Recherche d'un élément dans un tableau non trié

1. Écrire une fonction prenant en paramètre une liste L et un objet a, et retournant en sortie un booléen traduisant la véracité de l'affirmation « L'élément a est un élément du tableau ».
2. Modifiez la fonction ci-dessus afin qu'elle retourne l'indice minimum de la liste correspondant à une occurrence de a.
3. Même question pour trouver le nombre d'occurrences de a dans la liste.
4. On suppose la liste L remplie de façon aléatoire et uniforme par des entiers de $[\![0, N-1]\!]$, et $a \in [\![0, N-1]\!]$. On suppose la taille du tableau égale à $n = 10000$, qu'on estime « assez grand ». En faisant une moyenne sur 1000 expériences, et en introduisant dans la fonction de la première question un compteur de passages dans la boucle, retrouver par la simulation le fait que la complexité moyenne est à peu près N lorsque n est grand par rapport à N.
5. Déterminer de même une valeur approchée de l'écart-type du nombre de passages dans la boucle.

Exercice 2 – Recherche d'un élément dans un tableau trié

1. Étant donné un tableau trié T et un élément a, écrire une fonction, basée sur le principe de la dichotomie, retournant un indice i tel que $T[i] = a$, s'il en existe un, et retournant `None` sinon.
2. En introduisant un compteur dans la fonction ci-dessus, et en supposant T, de longueur n, rempli aléatoirement d'entiers choisis uniformément dans $[\![0, 2n]\!]$ puis trié, et en choisissant a aléatoirement entre 0 et $2n$, estimer le nombre moyen de passages dans la boucle (on prendra n en paramètre, on effectuera 1000 expériences, en tirant pour chaque expérience un nouveau tableau aléatoire ; on ne tiendra pas compte évidemment du coût du tri du tableau, qui n'est pas l'objet de cette mesure).
3. Comment modifier la fonction de la question 1 pour obtenir la première occurence de a, si elle existe ? La dernière occurrence ?

Exercice 3 – (Recherche d'un mot dans un texte par la méthode naïve)

1. Écrire une fonction prenant en paramètre deux chaînes de caractères `texte` et `mot`, et recherchant la première occurence de `mot` dans `texte` (la fonction retournera l'indice de la première lettre de l'occurrence de `mot` dans `texte`). On ne s'autorisera que des comparaisons <u>*lettre à lettre*</u>.
2. Écrire une fonction créant aléatoirement une chaîne de caractère de longueur 1000, constituée de façon aléatoire et uniforme des lettres (minuscules) de l'alphabet.
3. Estimez la probabilité qu'une chaîne aléatoire du type de la question précédente contienne le mot `llg`

Exercice 4 – (Paradoxe de Walter Penney)

Le but de cet exercice est d'illustrer par la simulation le paradoxe de Walter Penney : dans une succession infinie de tirages à Pile ou Face avec une pièce équilibrée, le temps d'attente moyen de la première configuration PPF est le même que le temps d'attente moyen de la première configuration FPP. Cependant, il est beaucoup plus probable qu'une configuration PPF ait lieu avant la première configuration FPP que l'inverse.

1. (a) Écrire une fonction simulant une succession de tirages à Pile ou Face jusqu'à obtention du premier motif PPF, et retournant le nombre de tirages nécessaires.

   (b) En répétant un grand nombre de fois l'expérience précédente, estimer le nombre moyen de tirages nécessaires pour obtenir pour la première fois PPF

2. Mêmes questions pour la séquence FPP.

3. Écrire une fonction retournant 1 si des deux motifs PPF et FPP, le motif PPF apparaît en premier, et 0 sinon.

4. Estimer la probabilité que le motif PPF apparaisse pour la première fois avant le motif FPP.