DS 4

Informatique de tronc commun, première année

Julien Reichert

Exercice 1 : Écrire une fonction prenant en entrée une liste `l` et qui retourne le nombre d'inversions dans cette liste, c'est-à-dire le nombre de fois où à deux indices `i` et `j` on a `i < j` mais l'élément en position `i` strictement supérieur à l'élément en position `j`. La complexité peut être quelconque, inutile d'optimiser comme vu en option pour une raison évidente.

Par exemple, la fonction retourne `4` pour la liste `[2, 5, 4, 6, 6, 4]` car le `5` est strictement supérieur aux deux `4` à sa droite et les deux `6` sont strictement supérieurs au `4` tout à droite.

Exercice 2 : Écrire une fonction prenant en entrée une liste représentant une permutation et qui retourne la décomposition en cycles à supports disjoints de la permutation. Par principe, on assimilera ici $\mathcal{S}_n$ à l'ensemble des permutations de l'intervalle $[|0, n-1|]$. La décomposition en cycles sera donnée en tant que liste de listes, chaque liste matérialisant un cycle (l'ordre des cycles et le premier élément de chaque cycle n'a pas d'importance).

Par exemple, la fonction retourne `[[0, 2], [1, 3, 6, 5]]` pour la liste `[2, 3, 0, 6, 4, 1, 5]` car la permutation qui à `0` associe `2`, à `1` associe `3`, ..., à `6` associe `5` se décompose en les cycles (02) et (1365) en ne tenant pas compte de 4 qui est point fixe.

Exercice 3 : Écrire une fonction prenant en entrée une fonction `fct` représentant une fonction réelle continue f (on ne le vérifiera pas) et trois flottants `a` et `b` et `d` et qui retourne une approximation de $\int_a^b f$ par la méthode des rectangles au milieu avec un pas de `d`. Concrètement il s'agit d'additionner des rectangles d'épaisseur `d` (sauf éventuellement le dernier) et de hauteur l'image par `fct` de l'abscisse du centre du rectangle en question.

Exercice 4 : Nous allons ici nous intéresser aux puissances de la matrice d'adjacence d'un graphe orienté.

Question 4.1 : Montrer que si M est la matrice d'adjacence d'un graphe G, dont on supposera que les sommets sont les entiers entre 0 et un certain entier naturel non nul n exclu, $k \in \mathbb{N}$ et $N = M^k$, alors pour tous $0 \leq i, j \leq n$ la valeur $N_{i,j}$ correspond au nombre de chemins de taille k du sommet i au sommet j dans G.

Question 4.2 : Écrire une fonction prenant en entrée deux listes de listes `M` et `N` représentant des matrices carrées de même forme et qui retourne une liste de listes correspondant au produit matriciel de `M` par `N` (l'ordre n'importera pas car on s'en servira exclusivement pour des puissances d'une même matrice...).

Question 4.3 : En déduire une fonction prenant en entrée une liste de listes `M` et un entier naturel `k` et qui retourne une liste de listes représentant la puissance `k`-ième de la matrice carrée représentée par `M`. Inutile d'optimiser l'exponentiation là aussi.

Question 4.4 : Déterminer ce que fait la fonction suivante, en notant `puissance` la fonction de la question précédente et en supposant qu'elle a été écrite et qu'elle est correcte.

```
def fonction(M, seuil):
    rep = 0
    for k in range(seuil+1):
        N = puissance(M, k)
        for i in range(len(M)):
            rep += N[i][i]
    return rep
```

Question 4.5 : Déterminer la complexité de la fonction précédente. Commenter.