

I Autour des listes

Exercice 1

Ecrire une fonction `maximum(L)` où `L` est une liste d'entiers et qui retourne la valeur du maximum de `L`, l'indice de la première occurrence ainsi que le nombre d'occurrences de ce maximum.

Exercice 2

Ecrire une fonction `est_triee(L)` qui renvoie un booléen indiquant si la liste `L` est triée dans l'ordre croissant.

Question bonus : comment pourrait-on modifier cette fonction pour prendre en compte l'ordre de tri ? On veut obtenir `est_triee(L, ordre)` où `ordre` vaudra 1 pour l'ordre croissant et `-1` pour l'ordre décroissant.

Exercice 3 (Bonus)

Ecrire une fonction `distance_max(L)` qui retourne la plus grande distance entre deux éléments de `L`.

Evaluer la complexité.

II Arithmétique

Exercice 4

Implémenter la fonction `pgcd(a, b)` qui retourne le pgcd de `a` et `b` en utilisant l'algorithme d'Euclide. On pourra en donner au choix une version itérative ou une version récursive.

Exercice 5

Ecrire une fonction `est_premier(n)` qui renvoie le booléen indiquant si `n` est un entier premier.

Exercice 6

L'algorithme du crible d'Eratosthène consiste à partir de la liste des entiers de 2 à `n`. On élimine ensuite de la liste tous les multiples de 2 puis de 3 et plus généralement du premier nombre restant encore dans la liste dont les multiples n'ont pas été éliminés.

Ecrire une fonction `eratosthene(n)` qui renvoie la liste de tous les nombres premiers compris entre 2 et `n` (sans utiliser l'exercice précédent).

III Algorithmes numériques

Exercice 7

Trouver `n` tel que $\sum_{k=1}^n \frac{1}{k} > 10$.

Exercice 8

Etant donné deux listes `T` et `L` (de même longueur) qui représentent respectivement des temps et les valeurs d'une certaine fonction `f` en ces temps, on cherche les points d'annulation de `f`.

On détecte ces annulations via la méthode suivante. Notons $T = [t_0, \dots, t_{n-1}]$. `f` change de signe entre `ti` et `ti+1` si et seulement si `f(ti)` et `f(ti+1)` (deux valeurs consécutives dans `L`) n'ont pas le même signe. Pour donner une approximation du temps d'annulation, on considère que `f` est affine sur $[t_i, t_{i+1}]$.

Ecrire une fonction `annulations(T, L)` qui renvoie la liste des temps où `f` s'annule.

IV Chaînes de caractère

Exercice 9

Ecrire une fonction `occurences_sous_chaine(motif, chaine)` qui renvoie le nombre d'occurrences de la chaîne `motif` dans `chaine`.

Par exemple pour le motif `'aba'` et la chaîne `'aababa'` on renvoie 2 (les occurrences n'ont pas à être disjointes).

Exercice 10

Ecrire une fonction `frequence_max(chaine)` qui renvoie la lettre qui apparaît le plus souvent dans la chaîne donnée. On considère que toutes les lettres sont en minuscule et que la chaîne ne comprend pas de caractère spécial.

Exercice 11

Ecrire une fonction `anagramme(mot1, mot2)` qui renvoie le booléen indiquant si les deux mots sont anagramme l'un de l'autre (sont composés des mêmes lettres, pas forcément dans le même ordre).