
TD 5 : PARCOURS DE LISTES, PREMIER TRI

Exercice 1 (Chercher dans une liste)

Écrire une fonction `cherche(L,x)` qui prend en entrée une liste `L` et un élément `x` et renvoie l'indice de la première occurrence de `x` dans `L` (ou `False` s'il n'y a pas).

La tester avec $x = 55$ la liste L constituée des 20 premiers entiers suivants : le premier est 1, le suivant s'obtient en ajoutant 2^2 au premier, le suivant en ajoutant 3^2 au second, etc

Exercice 2 (La recherche du maximum)

Écrire une fonction `maxi(L)` qui prend en entrée une liste `L` et renvoie le maximum de la liste. La tester.

Exercice 3 (Tri par comptage)

Dans cet exercice, on cherche à trier une liste d'entier dans l'ordre croissant.

On suppose ici que tous les termes de la liste sont des entiers naturels inférieurs ou égaux à un entier m fixé.

Principe : On compte le nombre d'itérations de chacun des entiers possibles dans la liste, puis on reconstruit la liste triée.

1. Écrire une fonction `nb_iter(L,k)` qui prend comme paramètres une liste `L` et un entier naturel `k` inférieur ou égal à m et renvoie le nombre d'itération de k dans la liste.
2. En déduire une fonction `tri_compt(L,m)` qui prend en entrée une liste L dont tous les termes de la liste sont des entiers naturels inférieurs ou égaux à un entier m fixé et renvoie la liste triée dans l'ordre croissant à l'aide de la méthode ci-dessus.
3. Modifier cette fonction pour qu'elle affiche un message d'erreur si la liste contient un entier qui n'appartient pas à $[0, m]$ et renvoie `None`.

Exercice 4 (Défi)

Écrire une fonction de tri qui prend en entrée une liste de points du plan (chacun étant représenté par une liste de deux points), dont on suppose ici que tous les termes de la liste sont des couples entiers naturels inférieurs ou égaux à un entier m fixé, et renvoie la liste des point triés dans l'ordre de leur distance spéciale à l'origine : la distance spéciale de (a, b) à l'origine est définie par $|a + b|$.