TP 5 bis IPT

Exercice 1 - Deux alorithmes de tri.

Les deux questions sont indépendantes.

1. Dans cette question, on veut trier une liste uniquement composée de 0 et de 1. L'astuce est qu'il suffit de compter le nombre de 0 (et de 1) puis de recréer une liste triée avec le bon nombre de 0 (et de 1).

Proposer une fonction $tri\_liste\_binaire(L)$ qui effectue ce tri, où la liste L est uniquement composée de 0 et de 1.

Quelle est la complexité de votre fonction ?

2. On rappelle l'algorithme du tri par insertion (tri des joueurs de cartes) :

- Le premier élément constitue le point de départ pour construire une liste triée.
- On insère l'élément suivant (le second pour commencer) dans la partie de la liste déjà triée à sa gauche.
- Recommencer le processus jusqu'à ce qu'il n'y ait plus d'élément de la liste initiale à insérer.

Proposer une fonction $tri\_insertion(L)$ qui trie une liste L dans l'ordre croissant.

La fonction combinera deux boucles, par exemple (mais ce n'est pas imposé), une boucle for et une boucle while .

Exercice 2 - Programmer des polynômes.

Un polynômes $P = \sum_{k=0}^{n} a_k X^k$ est codé par la liste de ses coefficients $L = [a_0, \dots, a_n]$. Par exemple, le polynôme $P = 1 + 2X - X^2$ est codé par la liste $L = [1, 2, -1]$.

1. Quelle commande utiliser pour trouver le degré d'un polynôme à partir de la liste L de ses coefficients ?

2. Etant donnée une liste L et un entier n, écrire une fonction $etendre(L, n)$ qui, lorsque n est plus grand que la longueur de L, renvoie une liste de n éléments, formée de ceux de L, puis complétée par des 0. Si n est inférieur ou égal à la longueur de L, on renverra L tel quel.

3. Etant donnés deux polynômes codés par les listes L_1 et L_2, écrire une fonction $somme(L_1, L_2)$ qui crée la liste des sommes des coefficients. On prendra soin de prendre en compte les difficultés si les listes L_1 et L_2 ne sont pas de mêmes longueurs. Par exemple, $somme([1, 1], [1, 0, 3])$ doit renvoyer $[2, 1, 3]$. On pourra utiliser la fonction précédente.

4. (Evaluation récursive). Etant donné un réel x et une liste $L = [a_0, \dots, a_n]$, on souhaite évaluer le polynôme associé en x, c'est-à-dire calculer $\sum_{k=0}^{n} a_k x^k$. Une astuce est une évaluation récursive, basée sur la formule : $P(x) = a_0 + x \times \sum_{k=0}^{n-1} a_{k+1} x^k$. Proposer une fonction $evaluation\_recursive(L, x)$ correspondant à cette idée. On rappelle que la commande $L[i :]$ extrait la sous-liste composée des éléments de la liste L à partir du i-ième (inclus).

5. On rappelle que le produit de deux polynômes $P = \sum_{k=0}^{n} a_k X^k$ et $Q = \sum_{k=0}^{m} b_k X^k$ est donné par $PQ = \sum_{k=0}^{n+m} c_k X_k$, avec $c_k = \sum_{i=0}^{k} a_i b_{k-i}$.

Ecrire une fonction $produit(L_1, L_2)$ qui renvoie la liste associée au produit des polyômes associés à L_1 et L_2.