

Révision 2 Info

Lundi

1. Écrire une fonction `TousPairs(L)` qui renvoie `True` si tous les termes de la liste d'entiers `L` sont pairs et `False` sinon.
2. Écrire une fonction `ExistePair(L)` qui renvoie `True` s'il existe au moins un élément pair de la liste d'entiers `L` et `False` sinon.

Mardi

1. Écrire une fonction `EstTrieé(L)` qui renvoie `True` si la liste de nombres `L` est triée par ordre croissant et `False` sinon.
2. Quelle est la complexité de cette fonction ?

Mercredi

1. Écrire une fonction `Doublons(L)` qui renvoie `True` si la liste ne contient aucun élément en double et `False` sinon. Pour cela, on pourra comparer `L[i]` et `L[j]` avec $i < j$.
2. Quelle est la complexité de cette fonction ?
3. Écrire `Doublons2(L)` qui renvoie `True` si la liste ne contient aucun élément en double et `False` en rentrant les éléments de `L` dans un dictionnaire.
4. Quelle est la complexité de cette fonction ?

Jeudi

1. Écrire une fonction `EstPremier(n)` qui renvoie `True` si `n` est premier et `False` sinon.
2. Quelle est la complexité de cette fonction ?
3. Si la complexité est linéaire, écrire une version dont la complexité est en $\mathcal{O}(\sqrt{n})$.

Vendredi

Soit $(u_n)_{n \in \mathbb{N}}$ définie par $u_0 = 2$ et par pour tout $n \in \mathbb{N}$, $u_{n+1} = \frac{5u_n + 3}{2u_n + 1}$

1. Écrire une fonction récursive `Suite(n)` qui renvoie u_n . Attention, il ne faudra faire qu'un seul appel récursif dans la fonction `Suite(n)`.
2. Écrire une fonction itérative `Suite2(n)` qui renvoie u_n .
3. Écrire une fonction `Affiche(n)` qui affiche sur un graphe les termes de la suite de 0 à n inclus.
4. Écrire une fonction `Seuil(n,s)` qui renvoie le plus petit n tel que $u_n > s$ avec $n \leq 10^6$ ou bien qui renvoie `False` dans le cas où pour tout $n \in \llbracket 0; 10^6 \rrbracket$, $u_n \leq s$.

Retrouvez ici, au fur et à mesure, les résultats!