

Licence STS 3^{ème} Année
Informatique
Aide à la Détection d'Erreurs

FEUILLE DE TD n°3

Écrire une fonction qui, étant donnée une liste d'entiers, renvoie une liste constituée des mêmes éléments mais triés en ordre croissant. Sa complexité en nombre de comparaisons devra être en $\theta(N \times \log_2(N))$, où N est la longueur de la liste.

Vous devrez procéder suivant les 6 étapes vues en cours (cf. recherche dichotomique), y compris la vérification de la complexité.

ATTENTION : pour obtenir une solution raisonnable il est nécessaire de décomposer le problème en sous-problèmes. Il y aura donc plusieurs algorithmes à fournir (chacun devant suivre les 6 étapes vues en cours).

On donne ci-dessous les notations algébriques afin de pouvoir décrire les idées et leur justification, puis les fonctions primitives de gestion des listes pour l'écriture des algorithmes.

Notations algébriques :

Soient L, L' deux listes d'entiers et e_i des entiers.

On note $L = \langle e_1, e_2, \dots, e_N \rangle$ la liste constituée des N éléments e_1, e_2, \dots, e_N dans cet ordre (attention deux éléments distincts peuvent avoir même valeur).

Si L est vide, on note $L = \langle \rangle$.

Si L est non vide (L contient N éléments avec $N > 0$), on note aussi $L = (e_1, L')$, où $L' = \langle e_2, \dots, e_N \rangle$.

Si L contient N éléments ($N > 0$) et si $i \in \{1, \dots, N\}$, on note aussi $\text{Elt}(L, i)$ le $i^{\text{ème}}$ élément de L (i.e. e_i)

Primitives sur les listes :

Soit `ListeEntier` le type liste d'entiers.

- Fonction avec résultat Booléen **TestListeVide(ListeEntier L)** qui retourne vrai ssi L est une liste d'entiers vide.
- Fonction avec résultat ListeEntier **ListeVide()** qui retourne une liste d'entiers vide.
- Fonction avec résultat ListeEntier **Cons(entier X, ListeEntier L)** qui retourne une liste d'entiers ayant comme premier entier X suivi des entiers de L .
- Fonction avec résultat Entier **Premier(ListeEntier L)** qui retourne le premier entier de L , ATTENTION : précondition : L doit être non vide.
- Fonction avec résultat ListeEntier **Suite(ListeEntier L)** qui retourne la liste constituée des entiers de L sans son premier entier, ATTENTION : précondition : L doit être non vide.

Toutes ces primitives ont une complexité en temps constante.

N.B. Les primitives appellent « naturellement » une approche récursive pour l'écriture des algorithmes. Une approche itérative compliquerait cette écriture.