Travaux Dirigés d'algorithmique n°7

Cours d'Informatique de Deuxième Année

—DEUG Sciences S3-MI—

Tri de listes chaînées

Nous nous intéressons ici à un algorithme de tri particulier, le `tri-fusion`, que nous appliquons aux listes simplement chaînées d'entiers.

Notons les listes en énumérant les éléments entre parenthèses.
Soit $l = (x_1, x_2, \ldots, x_n)$ une liste. L'idée du `tri-fusion` est la suivante :

- Si la liste l est vide ou ne contient qu'un seul élément, elle est déjà triée.
- Sinon
  - Couper la liste l en deux parties contenant chacune $n/2$ éléments.
  - Trier, par la même méthode, chacune des deux parties.
  - Fusionner les deux parties triées.

On veut implémenter en C le `tri-fusion` lorsque la liste est une liste chaînée d'entiers. Aucune des fonctions écrites ne devra faire d'allocation dynamique (autrement dit, les éléments de la liste ne doivent pas être recopiés, mais triés "sur place").
Vous devrez évaluer la complexité de chaque fonction. Pour cela :

▸ **Exercice 1.** *Définir un type, appelé* `Liste`*, pour représenter une listes chaînée d'entiers.*

▸ **Exercice 2.** *Écrire une fonction qui réalise la fusion de deux listes triées sans créer de nouvelles cellules. On rappelle que si* `f` *et* `g` *sont deux listes triées ayant respectivement n_1 et n_2 éléments, la fusion de* `f` *et* `g` *est la liste triée contenant les $n_1 + n_2$ éléments de* `f` *et* `g`*.*

▸ **Exercice 3.** *Écrire une fonction qui prend en argument une liste l et qui retourne un pointeur sur la cellule du milieu de l; c'est-à-dire, la fonction doit retourner* `NULL` *si l est vide, et un pointeur sur le $\lceil n/2 \rceil$ième élément de l sinon, où n est le nombre d'éléments de la liste l. En déduire une fonction qui coupe une liste l en deux listes de taille respectivement $\lceil n/2 \rceil$ et $\lfloor n/2 \rfloor$.*

▸ **Exercice 4.** *Déduire, des fonctions précédentes, une fonction qui implémente le* `tri-fusion` *d'une liste chaînée.*