

E.D. NFP 136 n°3

Thème : Listes

Exercice 1 Inversion d'une liste contiguë

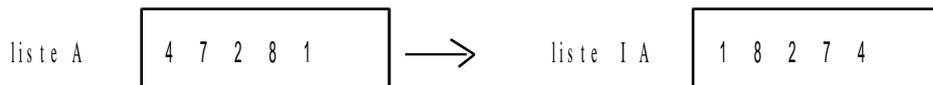
Proposer un algorithme pour inverser l'ordre des éléments dans une liste contiguë.
Ecrire la méthode Java associée, puis calculer sa complexité en temps et en espace.

Exercice 2 Inversion d'une liste chaînée (version itérative)

Question 1

Ecrire en Java une méthode `inverser` de la classe `Liste` (vue en cours) qui, étant donnée une liste non vide, renvoie une liste dans laquelle l'ordre des éléments a été inversé. On utilisera uniquement les 4 opérations de base de la classe `Liste` :

- D'abord, sans limitation de l'espace mémoire utilisé,
- Puis, en réduisant au maximum l'espace mémoire utilisé.



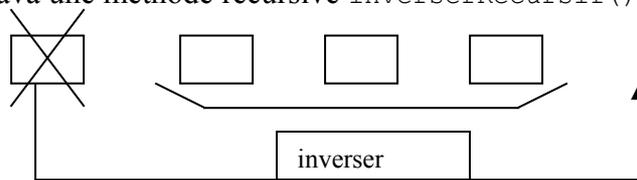
Question 2

Calculer la complexité en temps et en espace de ces deux méthodes.

Exercice 3 Inversion récursive d'une liste chaînée

On veut réaliser ici l'inversion d'une liste chaînée par une méthode récursive. On s'autorise pour cela à utiliser les 4 opérations de base de la classe `Liste`, ainsi que la méthode `insererEnQueue`, vue en cours.

Ecrire en Java une méthode récursive `inverserRecurusif()`, qui renvoie une liste.



Calculer ensuite la complexité en temps de cette méthode.

Exercice 4 Insertion après un élément donné dans une liste

Ecrire en Java une méthode de la classe `Liste` qui, étant donnée une liste non vide, permet d'ajouter un élément `x` donné juste après la première occurrence d'un élément `y` donné dans cette liste. (Le type de retour de cette méthode sera `void`.)