

TD : Structures de données

1 Listes

1. Implémenter une définition récursive d'une structure de liste simplement chaînée pour des éléments `int`. Tester.
2. Implémenter une méthode qui insère un élément au *début* d'une liste simplement chaînée. Supposer que la liste n'a **pas** de sentinelle de tête. À la place, utiliser une variable `head` qui référence le premier nœud de la liste.
3. Proposer et implémenter un algorithme pour trouver l'avant-dernier nœud dans une liste simplement chaînée où le dernier élément est repéré par sa référence nulle au suivant.
4. Décrire, puis implémenter un algorithme pour inverser les éléments d'une liste simplement chaînée, en utilisant seulement une quantité **constante** d'espace supplémentaire, et sans utiliser de récursivité.
5. Implémenter une méthode pour concaténer 2 listes doublement chaînées L et M , avec sentinelles de tête et queue, en une liste unique L' .
6. Décrire en détail, puis implémenter comment échanger 2 nœuds x et y (et pas seulement leurs contenus) dans une liste simplement chaînée, étant données les références vers x et y , et seulement ces références.
7. Répéter l'exercice précédent dans le cas d'une liste doublement chaînée.
8. Lequel des 2 algorithmes précédents est le plus rapide ?
9. Écrire une méthode qui compte le nombre de nœuds dans une liste chaînée circulaire.
10. Implémenter une méthode non-récursive de recherche du milieu d'une liste doublement chaînée avec sentinelles de tête et de queue, en ne procédant que par parcours des liens. Attention ! Cette méthode ne peut **pas** utiliser de compteur.
11. Étant donnée 1 liste circulaire contenant un nombre *pair* de nœuds, écrire une méthode qui divise la liste en 2 listes circulaires, chacune de la moitié de la taille initiale.

1.1 Listes génériques

2 Arbres

1. Implémenter une structure d'arbre d'éléments génériques.
2. Implémenter une méthode pour chacun des types de parcours d'arbre suivants : préfixé, infixé, postfixé.
3. Quelle est la valeur de l'expression arithmétique représentée par l'arbre illustré dans la figure suivante ?
4. Écrire un programme Java qui prend une expression arithmétique en entrée, et produit l'arbre de cette expression en sortie.