

Liste doublement chaînée

Table of Contents

1. Énoncé du problème
 2. Travail demandé
-

1. Énoncé du problème

On veut créer un type `Dble_Linked_List` permettant de représenter une *liste doublement chaînée*, c'est-à-dire une suite de *cellules* chaînées entre elles...

Une cellule contient donc 3 informations :

- un champ `valeur` (les valeurs contenues dans les cellules sont des entiers)
- un champ `suivant` "pointant" vers la cellule suivante
- un champ `precedent` "pointant" vers la cellule précédente



La liste n'a donc besoin que d'un "pointeur" vers la première cellule du chaînage (ce qui ne veut pas dire qu'il n'est pas utile d'y stocker d'autres informations).

2. Travail demandé

1. Écrire la classe `Dble_Linked_List` implémentant cette structure de données, ainsi que le type `Cellule`. Ces deux classes doivent redéfinir `__str__`.
2. Écrire les méthodes `cons` (ajout au début), `append` (ajout à la fin), `search` (rechercher la première occurrence), `remove` (supprimer la première occurrence), `to_list`, `from_list`, `card` (nombre d'éléments), `is_empty`, `concat` (concaténation de deux listes doublement chaînées, sans passer par `to_list` et `from_list`).



vous pouvez également redéfinir certains opérateurs (par exemple, `+` pourrait être un synonyme de `concat`)

Last updated 2021-12-13 15:19:27 +0100