5.3 Implémentation en Python des listes chainées

NSI Tle - JB Duthoit

5.3.1 Implémenter une liste chainée avec les classes

Une façon d'utiliser des listes chaînées avec Python, est d'utiliser une classe :

```
class Cellule:
"""une cellule d'une liste chaînée"""
        def __init__(self, v, s):
                self.valeur = v
                self.suivante = s
```

Ainsi la liste 1,2,3 est possible avec l'instruction :

```
liste = Cellule(1, Cellule(2, Cellule(3, None)))
```

Remarque

None remplace $\perp$.

Définition

Une liste chaînée est une structure de données pour représenter une séquence finie d'éléments. Chaque élément est contenu dans une cellule, qui fournit par ailleurs un moyen d'accéder à la cellule suivante. Les opérations sur les listes chaînées se programment sous la forme de parcours qui suivent ces liaisons, en utilisant une fonction récursive ou une boucle

5.3.2 Implémenter avec les tableaux ou tuples

Plutôt qu'un objet de la classe Cellule, on pourrait utiliser un couple, et dans ce cas écrire (1,(2,(3,None))).

On peut aussi encore un tableau à deux éléments, et dans ce cas écrire [1,[2,[3,None]]].

Une liste chaînée peut être encore être représentée par deux tableaux, l'un (contenu) contenant des valeurs et l'autre (suivant) contenant des indices. Le chaînage sera effectué de la façon suivante : l'élément suivant contenu[k] aura suivant[k] comme indice dans le tableau contenu.

	contenu	suivant
0	'2'	1
1	'3'	-1
2	'1'	0

Exercice 5.62
Ecrire une fonction listeN(n) qui reçoit un argument entier n et qui renvoie la liste chaînée des entiers 1,2,3,...,n dans cet ordre. Si n = 0, la liste renvoyée est vide.

Exercice 5.63
Ecrire une fonction longueur(lst) qui reçoit pour argument une liste chaînées et qui retourne la longueur de la liste chaînées.

1. L'écrire avec une boucle While
2. L'écrire comme fonction récursive

Exercice 5.64
Ecrire une fonction affiche_ liste(lst) qui reçoit en argument une liste chaînée et qui affiche, en utilisant print, tous les éléments de la liste lst, séparés par des espaces, suivies d'un retour chariot.

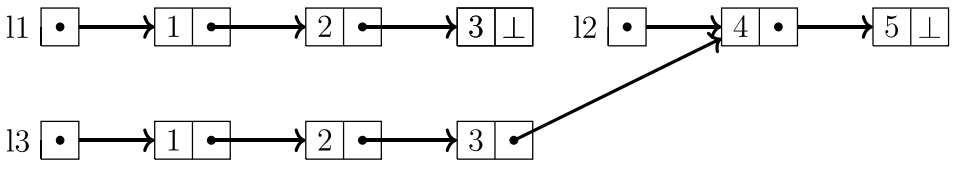
1. L'écrire avec une boucle While
2. L'écrire comme fonction récursive

Exercice 5.65
Ecrire une fonction n_ ieme_ element(lst) de paramètre lst une liste chaînée et qui renvoie le n-ième élément de la liste

1. L'écrire de façon récursive
2. L'écrire avec une boucle While

Exercice 5.66
On souhaite maintenant mettre bout à bout deux listes. On appelle cela une concaténation. Ecrire une fonction concatener qui reçoit deux listes en arguments et revoie une troisième liste contenant la concaténation.

```
def concatenerr(l1,l2):
    ''' concatène l1 et l2 ,
    sous la forme d'une nouvelle liste
    si l1 est vide, alors on renvoie l2
    sinon la concaténation est obtenue en concaténant la tête de l1 et la contenation du reste de l
    l1=Cellule(1,Cellule(2,Ceellule(3,None)))
    l2= Cellule(4,Cellule(5,None))
    l3 = concatener(l1,l2)
    '''
```



Remarque

On voit que les cellules de l1 ont été dupliquée tandis que les cellules de l2 partagées.Les cellules l2 permettent de constituer la liste l2 et la fin de la liste l3.Une alternative consisterait à dupliquer également les cellules de l2, mais ceci n'est pas forcément nécessaire.