

INFO-F-103 — Algorithmique 1

Devoir 1 – La magie de l’encapsulation

1 Énoncé

Un avantage des types de données abstraits est de pouvoir considérer l’instance d’une classe comme une boîte noire ayant un comportement défini par un ensemble de méthodes. Ainsi, lors de l’utilisation de l’objet, on ne doit aucunement se soucier de la façon dont ses données sont stockées, représentées ou manipulées.

Dans le premier devoir, nous avons créé une classe `SortedList` représentant une liste triée. Du point de vue du fichier `tests.py`, rien ne permet de deviner qu’on a utilisé la structure de données Skip list et pas une autre, on dispose simplement d’une classe permettant de faire des opérations d’insertion, de suppression et de recherche dans un ensemble d’éléments triés par valeur.

Un moyen plus simple que la Skip list pour maintenir une liste triée est d’utiliser une liste liée classique et de faire en sorte que la méthode d’insertion place directement l’élément au bon endroit dans la chaîne de références. Par exemple considérons la liste suivante:

```
HEAD -> 2 -> 3 -> 8 -> 15 -> 26
```

L’insertion de la valeur 12 se fait en parcourant la liste depuis la tête jusqu’au bon endroit puis en procédant à une insertion classique. Résultat:

```
HEAD -> 2 -> 3 -> 8 -> 12 -> 15 -> 26
```

La méthode de recherche (et par extension, de suppression) est légèrement différente de celle d’une liste non triée car elle peut s’arrêter dès qu’elle atteint un élément plus grand que celui recherché et ne doit pas explorer plus loin.

Dans ce deuxième devoir, nous vous demandons de créer une classe `SortedList` utilisant cette technique dont le squelette est le suivant:

```
class SortedList:

    def __init__(self):

    def insert(self, value):

    def remove(self, value):

    def search(self, value):

    def __iter__(self):
        current = self.head.get_next()
        while current is not None:
            yield current.get_value()
            current = current.get_next()
```

ainsi qu'une classe `Node`

```
class Node:

    def __init__(self, value, next):

    def get_next(self):

    def set_next(self, next):

    def get_value(self):

    def set_value(self, value):
```

En plaçant ces deux classes dans un fichier `SortedList.py`, il faut que les tests du premier devoir passent toujours sans qu'aucune modification ne soit faite au fichier `tests.py`.

2 Consignes

La version électronique est à remettre sur l'Université Virtuelle sous forme d'un fichier Python nommé `netID.py` (exemple: `fragerar.py`) contenant les classes `Node` et `SortedList` respectant les squelettes de code présentés plus haut.

La version papier sera à remettre à votre assistant lors de la séance d'exercices du vendredi 19 février 2016. Pour toute question → fragerar@ulb.ac.be