Universidade Europeia
LAUREATE INTERNATIONAL UNIVERSITIES

Folha Prática 5

ALGORITMOS E ESTRUTURA DE DADOS

Ana Mafalda Martins

2018/2019

Listas Ligadas (*Linked Lists*)

1. Implemente a classe *Node* e os métodos *__init__*, *isEmpty()*, *size()*, *add(item)*, *remove(item)* e *search(item)* da classe *LinkedList*, tal como feito na aula teórica.

2. Implemente o método *__str__* na classe *LinkedList* de modo a que quando se faça o *print* de uma lista ligada o output seja feito como as listas (*lists*) do Python (com colchetes).

   Exemplo:

```
my_list = LinkedList()
my_list.add(17)
my_list.add(93)
my_list.add(26)
my_list.add(54)
print(my_list)
[ 54, 26, 93, 17 ]
```

3. Na classe *LinkedList*, na implementação do método *size*, contamos o número de nós na lista. Uma estratégia alternativa seria armazenar o número de nós na lista como um dado adicional, além do cabeçalho (*head*) da lista. Modifique a classe *LinkedList* para incluir essa informação e reescreva o método *size*.

4. Altere o método *remove(item)*, para que ele funcione corretamente no caso em que *item* não esteja na lista.

   Nota: NÃO pode usar o método *search*.

5. Assumindo que a lista pode ter itens duplicados, o método *remove(item)*, tal como está implementado, remove da lista a primeira ocorrência do *item*. Altere esse método de modo a que sejam removidas todas as ocorrências do parâmetro *item* da lista.

6. Implemente, na classe *LinkedList*, os métodos restantes definidos no *LinkedList* TDA (Tipo Abstrato de Dados).

7. Implemente uma pilha (*Stack*) usando uma *LinkedList*.

8. Implemente uma fila (*Queue*) usando uma *LinkedList*.

9. Implemente o método *copy()* na classe *LinkedList*, que cria e devolve uma nova lista ligada que é uma cópia exata da lista (que o invoca).

10. Implemente o método *concatenate(b)* na classe *LinkedList*, que altera a lista concatenando-a à lista ligada *b*, ou seja, o último nó da lista deve passar a apontar para o primeiro nó da lista *b*.

11. A implementação da lista ligada (*LinkedList*) acima é chamada de lista simplesmente ligada (*singly linked list*) porque cada nó possui uma única referência para o próximo nó na sequência. Uma implementação alternativa é conhecida como uma lista duplamente ligada (*doubly linked list*). Nesta implementação, cada nó tem uma referência para o próximo nó (*next*), assim como uma referência o nó anterior (*previous*). A referência da cabeça (*head*) também contém duas referências, uma para o primeiro nó da lista e uma para o último. Codifique esta implementação em Python.