

# PUCRS – FACIN – Algoritmos e Estruturas de Dados I

## Exercícios de Lista Encadeada

Professora Isabel Harb Manssour

1. Acrescentar na classe *LinkedListOfInteger* um método que faça uma inserção ordenada dos elementos na lista (do menor para o maior). A assinatura deste método deve ser: **public void addIncreasingOrder(Integer element)**. Teste a sua implementação para este método na classe *App* e analise a caracterização *O* do seu tempo de execução.
2. Acrescentar na classe *LinkedListOfInteger* um **método não recursivo** que imprima o conteúdo da lista de trás para frente. Teste a sua implementação para este método na classe *App* e analise a caracterização *O* do seu tempo de execução.
3. Acrescentar na classe *LinkedListOfInteger* um **método recursivo** que imprima o conteúdo da lista de trás para frente. Teste a sua implementação para este método na classe *App* e compare com a solução anterior.
4. Acrescentar na classe *LinkedListOfInteger* o seguinte método: **public void reverse()**. Este método inverte o conteúdo da lista. Teste a sua implementação para este método na classe *App* e analise a caracterização *O* do seu tempo de execução.
5. Crie duas instâncias de *ListArrayOfInteger* na classe *App* e as inicialize com 100 valores aleatórios (entre 0 e 30). Depois implemente um algoritmo para descobrir e mostrar qual o maior valor que está armazenado nas duas listas simultaneamente. Para gerar números randômicos use a classe *Random* (*Random r = new Random(); r.nextInt(30);*). No exemplo abaixo, com listas de 5 elementos, o maior valor armazenado nas duas é 07.  
L1: 04,12,07,30,20  
L2: 50,07,04,21,33
6. Implemente um método que percorre a lista e retira elementos repetidos, deixando apenas uma ocorrência de cada elemento. A assinatura deste método deve ser: **public void unique()**. Teste a sua implementação para este método na classe *App* e analise a caracterização *O* do seu tempo de execução.