

Prova de Introdução a Ciência da Computação II – Turma 94
Prof. Eduardo C. Xavier

Questão 1 (3 pontos) – Suponha que estamos trabalhando com uma lista ligada simples com sentinela (classe ListaLigada), onde cada nó é da seguinte forma

class No{ No prox=null; int valor; }	e a lista tem a seguinte definição	class ListaLigada{ No cabeca; public ListaLigada(){ cabeca = new No(); } }
-----------------------------------------------	------------------------------------	-------------------------------------------------------------------------------------------

Nas questões abaixo você não pode pressupor a existência de métodos insere, remove etc, pois estes não estão definidos na classe.

a) Escreva um método em Java que recebe uma lista ligada e devolva uma NOVA lista (a lista deve ter seus próprios nós) que é o inverso da lista recebido por parâmetro.

b) Escreva um método que receba uma lista ligada por parâmetro e um valor inteiro **i**. O método deve retornar uma NOVA lista ligada (com novos nós) que é uma cópia da lista passada por parâmetro sem os **i** primeiros elementos. Se **i** for maior do que a quantidade de nós, o método retorna uma lista vazia.

Questão 2 (2 pontos) – Apresente uma definição recursiva para se achar simultaneamente os valores mínimo e máximo de um vetor de inteiros. Escreva um método recursivo em Java que implemente a sua definição recursiva.

Questão 3 (2,5 pontos) – Escreva um método em Java que recebe por parâmetro um vetor de inteiros, e posições **i** e **j** do vetor tal que **i** ≤ **j**. Assuma que existe uma variável **n** da classe que indica que de 0 até **n** há dados úteis no vetor. O método deve remover os valores do vetor da posição **i** até a posição **j** inclusive. A remoção consiste em um deslocamento para trás de dados no vetor dos elementos das posições **j+1** até **n**. Faça uma análise assintótica do tempo de execução do seu método. Assuma que os índices são válidos isto é $0 \leq i \leq j \leq n$.

Questão 4 (2,5 pontos) – Escreva um método em Java que receba um vetor **v** de inteiros por parâmetro e retorna um valor booleano indicando se o vetor está organizado de tal forma que $v[i+1] == v[i] + 2$, para cada **i** de **0** até **n-2**, ou seja, cada posição é o valor da posição anterior mais 2. Faça uma análise da complexidade de tempo do seu algoritmo.