Listas, Pilhas e Filas

1) Seja L uma lista simplesmente encadeada. Escreva um algoritmo que, percorrendo a lista uma única vez, constrói uma nova lista L' que:
   A. possui os valores de L em ordem inversa
   B. A nova lista L' possui a metade dos nós da lista original, onde o primeiro nó de L' contém a soma do primeiro nó de L com o último nó de L, o segundo nó de L' contém a soma do segundo nó de L com o penúltimo nó de L e assim por diante: L'= < L1+Ln, L2+Ln-1, L3+Ln-2, … , Ln/2 + Ln/2+1>, onde n é sempre par.

2) Escreva um algoritmo para reconhecer se uma dada palavra é um palíndromo. Considere que a palavra está contida em uma lista simplesmente encadeada, onde cada caractere está em um nó da lista.

3) Seja A uma matriz esparsa n x m.

a) Crie uma estrutura de dados que represente A e cujo espaço total seja O(k) em vez de O(mn), onde k é o número total de elementos não irrelevantes de A.

b) Faça um algoritmo para localizar um valor a_{ij} na estrutura acima.

c) Faça um algoritmo para computar A^2 utilizando a estrutura acima. Para tal, crie os algoritmos de inserção e busca na sua estrutura.

d) Qual a complexidade da sua solução para a letra (c)?

3.5) Caso não tenha pensado numa estrutura com duas listas (uma para colunas e uma para linhas), recomenda-se refazer o exercício acima com essa forma de estruturação.

4) Listas são usadas para representar números muito grandes (p.ex, com 1000 dígitos), uma vez que seria impossível representá-lo em máquinas de 64bits. Para representar inteiros grandes com listas, é usada uma representação em que cada dígito do inteiro é armazenado em um nó da lista. Considere duas listas encadeadas L1 e L2 representando números grandes (cada digito por nó). Faça um algoritmo que faça a soma de dois inteiros grandes e retorne a lista L3 = L1 + L2.

5) Para cada estrutura abaixo, implementa os métodos contido(K,L), inserir(K,L) e remover(K,L):

a) Lista duplamente encadeada: considere a inserção sempre no **final** da lista

b) Fila com lista simplesmente encadeada: considere que vc tem uma variável head e uma tail, onde head marca a cabeça e tail o ultimo nó na cauda.

c) Pilha com lista simplesmente encadeada

d) Lista duplamente encadeada circular

6) Verifique a complexidade de cada uma das 3 operações do exercício anterior para cada tipo de estrutura.

7) Implemente o algoritmo de contido() em uma skiplist considerando a estrutura abaixo.

```
struct No       { No *prox; No *desce, int x; }
struct SkipList { No *head; }
boolean contido(No *head, int x);
```

Obs: recomenda-se fazer o exercício 3.5 antes para ganhar familiaridade com uso de 2 listas.