MAC5710 - Estrutura de Dados e suas Aplicações

Primeiro semestre de 2009

Lista 1

1. Escreva uma função que inverte uma dada lista ligada. A lista resultante deve conter as mesmas células que a original, porém na ordem inversa. Não troque o conteúdo das células. Imagine que as células da lista dada estejam sendo referenciadas por uma outra estrutura de dados. Quanto tempo consome a sua função em relação ao número de elementos da lista?

2. Escreva uma função que ordena uma dada lista ligada. Novamente não troque o conteúdo das células. Apenas as rearranje de modo que, ao final, a lista se encontre ordenada. Quanto tempo consome a sua função em relação ao número de elementos da lista? Você consegue fazer melhor que isso?

3. Escreva uma função que busca um elemento numa dada lista ligada. Reescreva a sua função supondo que a lista está ordenada.

4. Escreva uma função que recebe duas listas ligadas ordenadas e devolve uma lista ordenada obtida da intercalação das duas listas. A lista resultante deve ser composta das células das duas listas originais. Não troque conteúdos de lugar. As duas listas de entrada serão destruídas no processo. Quanto tempo consume a sua função em relação ao comprimento das duas listas originais?

5. Escreva uma função que recebe uma lista duplamente ligada e remove todas as duplicações que ela contenha. Ou seja, a função deixa a lista com exatamente um exemplar de cada elemento. Utilize para isso a função busca que você escreveu no exercício 3 acima.

6. Implemente uma fila dupla com uma lista duplamente ligada com cabeça. Mostre as 6 funções de manipulação de fila dupla. Quanto tempo consome cada uma dessas operações?

7. Considere um conjunto armazenado em uma lista ligada simples. Escreva uma função para cada uma das operações abaixo:

   - `Pertence(A, x)` que verifica se `x` pertence ou não ao conjunto dado por `A`.
   - `União(A, B)` que recebe dois conjuntos, apontados por `A` e por `B`, e devolve a união deles, ou seja, um apontador para uma lista ligada que representa a união dos dois conjuntos. Os dois conjuntos devem ser mantidos intactos. Note que cada elemento do conjunto deve aparecer na lista uma única vez.
   - `Intersecção(A, B)` que recebe dois conjuntos, apontados por `A` e por `B`, e devolve (a lista que representa) a interseção entre eles. Novamente os dois conjuntos devem ser mantidos intactos.
   - `Diferença(A, B)` que recebe dois conjuntos `A` e `B` e devolve a diferença do conjunto `A` por `B`.

8. Considere a implementação de polinômios esparsos utilizando listas ligadas circulares com cabeça de lista, com os monômios não-nulos ordenados pelo expoente. Escreva uma função para cada uma das operações abaixo:

   - `Subtrai(p, q)` que recebe dois polinômios `p` e `q` e devolve o polinômio $p - q$.
   - `Valor(p, x)` que devolve o valor de `p` em `x`.
   - `Raiz(p, x)` que verifica se `x` é ou não uma raiz do polinômio `p`.