

Terceira (Super)Lista de Exercícios de Estrutura de Dados e Algoritmos 2/2006

1. Apresente um algoritmo ótimo para calcular o k -ésimo elemento de um conjunto quando existem valores repetidos. Justifique o *porquê* de seu algoritmo ser ótimo.
2. Escreva um algoritmo, e implemente em C, que leia uma seqüência (não ordenada) de N valores e diga qual valor (ou quais valores) aparece(m) mais vezes na seqüência. Qual o custo assintótico de seu algoritmo?
3. Como podemos simular uma fila usando duas pilhas? Suponha que os procedimentos de inserção, remoção e teste de pilha vazia já estão prontos. Escreva os procedimentos correspondentes de inserção e remoção de uma fila implementada por duas pilhas. Neste caso, qual o custo de inserção e remoção de um elemento nesta *fila*.
4. Escreva procedimentos para inserir e remover elementos numa fila utilizando uma lista encadeada simples. As operações de inserir e remover elementos da fila devem ter custo $\Theta(1)$.
5. Escreva os procedimentos de *inserção* e *remoção* de elementos numa *pilha* usando lista encadeada. Avalie o custo de cada operação.
6. Escreva um procedimento não recursivo, com tempo de execução $\Theta(n)$ que inverta uma lista encadeada simples de n elementos. Além do custo de armazenar os n elementos, o procedimento não deve gastar mais do que $O(1)$ para inverter a lista.
7. Seja E um vetor de N inteiros, o *somatório da subcadeia máxima* é o máximo da soma dos valores da subcadeias (elementos em posições consecutivas) do vetor. Se todos os elementos de E são negativos o somatório da subcadeia máxima é zero. Por exemplo, para a cadeia com elementos:
38, -62, 47, -33, 28, 13, -18, -468, 21, 12, -53, 25, o somatório da subcadeia máxima é 55 (posição 3 à 6). Escreva um algoritmo que calcula o somatório da subcadeia máxima do vetor E . Analise o tempo de execução do algoritmo proposto.

8. Seja **A** e **B** vetores de N inteiros cada um. Uma *subseqüência comum* de **A** e de **B** é uma seqüência que é subseqüência de **A** e é subseqüência **B**. A subseqüência não requer que os elementos sejam consecutivos. Por exemplo, se os valores de **A** são 5, 8, 6, 4, 7, 1, 3 e os valores de **B** são 4, 5, 6, 9, 7, 3, 2, a mais longa subseqüência comum é 5673. Apresente um algoritmo para achar a subseqüência comum mais longa de **A** e **B**. Analise os custos de execução no pior caso.
9. Escreva um programa que dado um texto (gabarito), e um ditado(texto a ser analisado) fornecidos em arquivos de entrada, alinha os trechos semelhantes e marca as diferenças.
 por exemplo:
 Artist=is=a=descriptive=term=applied=to=a=person...
 Artist=-s=a=descreption=te-m=ap-lied=to=a=persom...
 =====x=====x===xx===x====x=====x...
10. Dado um polígono P (descrito por seus vértices no plano R^2) e um ponto q , escreva um algoritmo que determina se o ponto esta dentro ou fora do polígono (atenção: o polígono pode não ser convexo). Qual a complexidade assintótica de seu algoritmo?
11. Árvores binária de busca.
 Resolva os seguintes exercícios do Cormen (2a edição): 12.3 – 3, 12.3 – 4, 12.3 – 5.
12. Escreva e implemente em linguagem C o Algoritmo de *Heapsort* para ordenar uma seqüência de n valores em ordem decrescente.

OBS: Data de entrega da lista: 14/11/2006.