

Primeira Avaliação a Distância

1. Para cada item abaixo, responda “certo” ou “errado”, justificando (meio ponto cada):
 - a. Se a complexidade de pior caso de um algoritmo A para um problema P for $O(f)$, então o número de passos que um algoritmo ótimo para P efetua, em qualquer caso, é igual a $O(f)$.
 - b. Se a complexidade de melhor caso de um algoritmo for $\Theta(f)$, então o número de passos que o algoritmo efetua, qualquer que seja a entrada, é $O(f)$.
2. (1,0) Considere uma sequência de elementos f_1, f_2, \dots, f_n , definida do seguinte modo: $f_1 = 0$, $f_2 = -1$, $f_3 = 1$, $f_j = f_{j-1} - f_{j-2} + f_{j-3}$ para $j > 3$. Elaborar algoritmos, um recursivo e outro não recursivo, para determinar o elemento f_n da sequência. Determinar a complexidade dos algoritmos.
3. (2,0) Escreva um algoritmo que inverte uma lista simplesmente encadeada com nó cabeça. Exemplo: se a lista contém os elementos 1, 3, 5, 9, 2, nesta ordem, então a lista resultante contém os elementos 2, 9, 5, 3, 1. Atenção: não é permitido o uso de estruturas auxiliares para realizar a inversão, como um vetor auxiliar ou uma pilha. Responda: sem o uso de estruturas auxiliares, é possível realizar a inversão em tempo inferior a $O(n^2)$?
4. (2,0) Faça uma tabela que liste as complexidades de melhor e pior caso dos algoritmos de busca, inserção e remoção, em listas sequenciais e simplesmente encadeadas, ordenadas e não ordenadas. Note que a tabela contém 12 entradas! Observação: as complexidades de inserção e remoção devem desconsiderar o tempo prévio para fazer a busca do elemento.
5. (2,0) Compare as complexidades dos métodos de ordenação vistos nas aulas quando o vetor de entrada já está ordenado crescentemente, supondo elementos distintos entre si. Repita o exercício quando o vetor de entrada vem dado em ordem decrescente.
6. (2,0) Considere uma lista sequencial L com n elementos distintos, e um vetor V com m elementos quaisquer. Deseja-se realizar a seguinte tarefa: para cada elemento x de V , deve-se verificar se x está em L . Se x não estiver em L , deve-se inseri-lo em L ; caso contrário, nada a fazer. Em qualquer caso, continua-se a tarefa para o próximo elemento de V . Elabore um algoritmo eficiente para resolver este problema. Procure elaborar um algoritmo com a melhor complexidade possível.