

Lista de Exercícios 3 (5 Pontos)

As questões a seguir requerem implementação em linguagem de Programação Java. Organize as questões de implementação em um Projeto com menu de opções para a seleção do algoritmo.

1) Dado um vetor com n números inteiros deseja-se reorganizar os elementos de forma que todos os números negativos precedam os não negativos e os zeros fiquem entre eles. Desenvolva e implemente em Java um algoritmo com divisão e conquista que reorganize esse vetor com ordem de complexidade $\theta(n \log n)$. Forneça um exemplo para ilustrar seu algoritmo.

2) Considere o jogo da velha. O jogo consiste em cada jogador colocar o valor X ou O numa posição da tabela 3X3. Um jogador ganha quando ele completa uma linha, coluna ou diagonal com o seu símbolo. Projete e implemente em Java uma estratégia vencedora, se houver, para o jogador que começa jogando utilizando uma abordagem *backtracking*.

3) Seja o problema: um ladrão que rouba uma loja encontra n itens: o i -ésimo item vale $v[i]$ reais e pesa $w[i]$ quilos, onde $v[i]$ e $w[i]$ são inteiros. Ele deseja levar uma carga tão valiosa quanto possível, mas pode carregar no máximo W quilos em sua mochila, para algum inteiro W . Cada item deve ser levado ou deixado para trás; o ladrão não pode levar uma quantidade fracionária de um item ou levar um item mais de uma vez. Proponha a implementação de algoritmos que resolva esse problema utilizando abordagem gulosa e programação dinâmica.

4) Apagando e Ganhando é um programa de auditório muito inteligente. Os participantes são selecionados por sorteio, e o apresentador escreve em uma lousa um número de N dígitos. O participante deve então apagar D dígitos do número que está na lousa. Os números que restarem equivalem ao prêmio em dinheiro que ele leva para casa!

Exemplo: Número na lousa: 1 2 3 1 2 3 9

Dígitos a serem apagados: 3

Total a ser levado para casa: 1 2 **3** 1 **2 3 9** = 3239!

Desenvolva um algoritmo guloso que resolva esse problema. Implemente em Java sua solução.

5) Utilizando a técnica de *backtracking* proponha a implementação de um algoritmo em Java que resolver o problema das N Rainhas, que consiste em colocar N rainhas em um tabuleiro de xadrez

de tamanho $N \times N$ de forma que as rainhas não se ataquem. O valor de N poderá ser informado pelo usuário.

6) Dada duas sequências X e Y , dizemos que uma sequência Z é uma subsequência comum de X e Y se Z é uma subsequência de X e de Y ao mesmo tempo.

Exemplo:

$X = \langle A, B, C, B, D, A, B \rangle$ e $Y = \langle B, D, C, A, B, A \rangle$ a sequência $\langle B, C, A \rangle$ é uma subsequência comum. A sequência $\langle B, C, B, A \rangle$ é uma subsequência comum mais longa (LCS – longest common subsequence).

a) Implemente em Java um algoritmo baseado em programação dinâmica que determine a subsequência comum mais longa de duas *strings* quaisquer X e Y .

b) Apresente a matriz final preenchida com valores processados pelo algoritmo proposto considerando o exemplo acima.

7) Suponha você possui uma lista A de tarefas a serem realizadas. Cada tarefa i possui uma duração t_i . Considerando que você dispõe de T unidades de tempo para realizar o máximo de tarefas:

Implemente um algoritmo guloso que determina quais tarefas serão realizadas no tempo T .

8) Implemente pelo menos dois algoritmos baseado em programação dinâmica que resolva um dos seguintes problemas: sequência de Fibonacci, Distância de edição/Levenshtein, problema do troco e Multiplicação de matrizes.

Questões teóricas/manuais
(Exclusivas para a Disciplina de Projeto e Análise de Algoritmos)

9) Apresente a complexidade no pior caso para os algoritmos implementados nas questões anteriores.

10) As técnicas de projeto de algoritmos são essenciais para que os desenvolvedores possam implementar software de qualidade. Essas técnicas descrevem os princípios que devem ser adotados para se projetar soluções algorítmicas para um dado problema. Entre as principais técnicas, destacam-se os projetos de algoritmos força bruta, transformar para conquistar, divisão e conquista, retrocesso (*backtracking*), programação dinâmica e algoritmos gulosos. Nesse contexto, apresente pelo menos duas características de cada uma das técnicas.

11) Descreva pelo menos duas características que classificam um problema computacional como *NP-Completo*.

Dica para a Prova III: treine a execução dos algoritmos baseados em programação dinâmica abordados nesta lista e nas aulas, a partir de exemplos variados.