

FCT/Unesp – Presidente Prudente
Projeto e Análise de Algoritmos
Prof. Danilo Medeiros Eler

Trabalho Prático 02
30/10/2019

Instruções de Envio

Enviar para **daniloelerunesp@gmail.com** (por favor, enviar **somente** para esse email)

No assunto do email você deve colocar: [PAA2019] Trabalho Prático 02.

No corpo do email você deve identificar o seu grupo, colocando o nome de cada integrante. Anexar o relatório com explicação dos problemas e das soluções, bem como um exemplo de execução (entrada e saída); código fonte; e, se houver, os dados utilizados pelos algoritmos (no caso de ele estar lendo de arquivo).

Importante: para o gmail não barrar executáveis (e.g., .exe, .jar), renomeie a extensão do arquivo **.zip** ou **.rar** para, respectivamente, **.renomearParazip** ou **.renomearPararar**.

Data máxima para envio

O trabalho deve ser enviado por email até o dia **01/12/2019**.

Especificações do trabalho prático

O trabalho pode ser desenvolvido em grupos de no máximo 3 pessoas.

Desenvolver programas para resolver os problemas listados abaixo, utilizando a técnica de projeto de algoritmos indicada. Além da implementação, deve ser elaborado um relatório com a descrição dos problemas e das soluções apresentadas, indicando a ordem de complexidade no pior caso. O relatório também deverá conter um exemplo de execução, mostrando a entrada e a saída de dados.

Os programas devem possuir uma interface intuitiva e de fácil uso, preferencialmente, interfaces gráficas. As interfaces em modo texto devem ser bem planejadas para não causar dúvidas e dificuldades para o usuário do programa. Isso será levado em consideração no momento da avaliação do trabalho.

Os problemas que fazem parte deste trabalho são:

1. Problema de Associação de Tarefas (*Assignment Problem*)
 - Utilizar tentativa e erro com *Branch and Bound*
2. Codificação de Huffman para compressão de um texto fornecido pelo usuário
 - Utilizar um algoritmo guloso
3. Problema da Mochila Fracionária (*Fractional Knapsack Problem*)
 - Utilizar um algoritmo guloso (pesquisar sobre o problema)
4. Problema da Mochila Booleana (mochila 0-1 – em inglês, *Knapsack Problem*)
 - Utilizar Programação Dinâmica
5. Problema da Subsequência Comum Máxima (*Longest Common Subsequence*)
 - Utilizar Programação Dinâmica