UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL REI
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
PROJETO E ANÁLISE DE ALGORITMOS

1º semestre de 2025
Professor: Leonardo Chaves Dutra da Rocha

Trabalho Prático 3
Data de Entrega: 26 de Junho 2025.

Este trabalho prático tem por objetivo exercitar conceitos e práticas dos algoritmos de processamento de caracteres e compressão de arquivos.

Definição do Problema

O objetivo deste trabalho é realizar experimentos em um conjunto de programas para recuperar ocorrências de padrões em arquivos constituídos de documentos comprimidos e não comprimidos. O sistema de programas recebe do usuário um texto contendo n caracteres, o padrão contendo m caracteres, o número de erros ($0 \leq k < m$), e imprime todas as ocorrências do padrão no texto. Dividimos esse trabalho em duas frentes

Parte 1: Busca Aproximada em Arquivos Não Comprimidos
Nesta parte do trabalho você deverá utilizar os seguintes algoritmos:

- Algoritmo programação dinâmica para casamento aproximado.
- Algoritmo Shift-And para casamento aproximado.

Compare o desempenho dos dois algoritmos para valores de k = 0,1,2,3. Considere o desempenho de duas maneiras diferentes (1) Medindo o número de comparaçõoes; e (2) Medindo o tempo de relógio.

Entrada e Saída
Nesse caso, o executável deverá se chamar tp4_parte1 e deverá receber três parâmetros: o algoritmo a ser utilizado (1 para programação dinâmica e 2 para Shift-And), um arquivo contento o texto onde deverão ser feitas as buscas e um arquivo contendo os padrões a serem procurados, um por linha. A saída deve conter, para cada padrão procurado, a lista de ocorrências do mesmo. Segue abaixo um exemplo:
Arquivo Texto:

```
Texto exemplo, texto tem palavras, palavras exercem fascínio.
```

Arquivo Padrões:

```
palavras
exemplo
```

Arquivo Saída

```
palavras 26 36
exemplo 7
```

Parte 2: Busca Exata em Arquivos Comprimidos
Nesta parte do trabalho você deverá comparar o desempenho do algoritmo de Boyer-Moore Horspool (BMH) para para arquivos comprimidos e não comprimidos. Para comprimir o arquivo você deve utilizar o código de Huffman com marcação descrito em Ziviani (2004, Seção 8.2). Um arquivo comprimido com esse algoritmo permite que o padrão de busca seja comprimido e buscado diretamente no arquivo comprimido.

Compare o desempenho do algoritmo BMH para arquivos comprimidos e não comprimidos. Considere o desempenho de duas maneiras diferentes (1) Medindo o número de comparaçõoes; e (2) Medindo o tempo de relógio.

Entrada e Saída

Nesse caso, o executável deverá se chamar tp4_parte2 e deverá receber dois parâmetros: um arquivo contento o texto onde deverão ser feitas as buscas e um arquivo contendo os padrões a serem procurados, um por linha. Para essa parte, em um primeiro momento a rotina de compressão de Huffman deverá ser chamada. A saída deve conter, para cada padrão procurado, a lista de ocorrências do mesmo. O mesmo exemplo da parte 1 pode ser considerado aqui.

Observações:

- O código fonte do trabalho deve ser submetido para compilação e execução em ambiente Linux, tendo como padrão os computadores dos laboratórios do DCOMP.
- Deve ser escrito na linguagem C (trabalhos implementados em outras linguagens como C++/Java/Python e outras não serão aceitos);
- As estruturas de dados devem ser alocadas dinamicamente e o código deve ser modularizado utilizando os arquivos .c .h.
- O utilitário Make deve ser utilizado para compilar o programa;
- A saída deve ser impressa no arquivo pedido seguindo estritamente o formato da especicação caso contrário o resultado será considerado errado;
- Faça seu código de forma legível

Avaliação

Deverão ser entregues:

- listagem das rotinas;
- documentação contendo:;
  - descrição das soluções e estruturas de dados utilizadas;
  - análise da complexidade das rotinas;
  - análise dos resultados obtidos.
  - a documentação não pode exceder 12 páginas.

Distribuição dos pontos:

- execução (E)
  execução correta: 80%
- estilo de programação
  código bem estruturado: 10%
  código legível: 10%
- documentação (D)
  comentários explicativos: 30%
  análise de complexidade: 10%
  análise de resultados: 60%

A nota final é calculada como a média harmônica entre execução (E) e documentação (D):

$$\frac{D * E}{\frac{D+E}{2}}$$