

A3 – Sistema de Gestão e Busca de Documentos Compressos com Índice Dinâmico

Desenvolva um sistema capaz de gerenciar, indexar, ordenar e buscar documentos de texto armazenados em disco. O sistema deve suportar a inserção, remoção, ordenação e compressão de documentos, além de oferecer buscas eficientes por palavras-chave. Os documentos devem ser armazenados de forma persistente em arquivos e indexados por múltiplas estruturas de dados, como árvores, tabelas hash e estruturas auxiliares para compressão e ordenação.

Você deve implementar:

- a. Estruturas dinâmicas de dados para gerenciar os documentos e seus índices.
- b. Algoritmos de ordenação eficientes e análise de complexidade dos mesmos.
- c. Compressão de documentos usando codificação de Huffman.
- d. Índice reverso persistente com B+Tree ou Hashing para busca por palavras.
- e. Suffix Array e LCP Array para buscas por padrões de texto.

Itens Avaliativos e Pontuação (Total: 40 pontos)

1. Cadastro e Armazenamento de Documentos (8 pts)

- 1.1.** Leitura e armazenamento de documentos em memória usando listas ligadas **(2 pts)**
- 1.2.** Persistência em arquivos com estrutura de índice BTree+ (pode ser simulada) **(3 pts)**
- 1.3.** Compressão com Huffman para salvar os textos no disco **(3 pts)**

2. Busca por Palavras-Chave com Índice de Relevância (10 pts)

- 2.1.** Implementar índice invertido usando HashMap com listas **(3 pts)**
- 2.2.** Usar Suffix Array + LCP para encontrar trechos com prefixo comum **(5 pts)**
- 2.3.** Classificar os resultados por frequência com Heap Sort **(2 pts)**

3. Árvores e Estruturas de Navegação (12 pts)

- 3.1.** Armazenar palavras ordenadas em AVL para buscas rápidas **(4 pts)**
- 3.2.** Usar Rubro-Negra para gerenciar usuários por nome **(4 pts)**
- 3.3.** Utilizar BTree para armazenar metadados dos documentos **(4 pts)**

4. Algoritmos de Ordenação (8 pts)

- 4.1.** Implementar ordenações: Selection, Merge, Quick e Heap **(4 pts)**
- 4.2.** Comparar a performance dos algoritmos com análise de tempo real **(2 pts)**
- 4.3.** Escolher o melhor algoritmo para cada cenário de ordenação (por tamanho, por data etc.) **(2 pts)**

5. Testes e Interface (7 pts)

- 5.1.** Criar interface em linha de comando (CLI) para as funcionalidades **(2 pts)**
- 5.2.** Criar testes com JUnit para pelo menos 5 componentes chave **(3 pts)**
- 5.3.** Usar recursão em ao menos uma estrutura (ex: travessia de árvore) **(2 pts)**

6. Complexidade e Desempenho (5 pts)

6.1. Documentar complexidade assintótica das principais operações (3 pts)

6.2. Medir tempo médio de execução e uso de memória (2 pts)







Telas e iteração da solução

Sugestão de telas para uma interface gráfica (GUI) ou adaptável para interface em terminal (CLI/TUI), pensando na organização do seu Sistema de Gestão e Busca de Documentos Compressos com Índice Dinâmico.

Telas Sugeridas

1. Tela Inicial / Menu Principal

Componentes:

- Título do sistema
- Menu de opções:
 -  [1] Cadastrar novo documento
 -  [2] Listar documentos cadastrados
 -  [3] Buscar por palavra-chave
 -  [4] Estatísticas e ordenação
 -  [5] Gerenciar índice ou compressão
 -  [0] Sair

Exemplo (CLI):

```
===== Sistema de Documentos Compressos com Índice Dinâmico =====
1 - Cadastrar novo documento
2 - Listar documentos cadastrados
3 - Buscar por palavra-chave
4 - Ver estatísticas e ordenações
5 - Gerenciar compressão e índices
0 - Sair
Escolha uma opção:
```

2. Tela de Cadastro de Documento

Componentes:

- Campo para nome do documento
- Área de texto para conteúdo
- Botões: [Salvar] [Cancelar]

Fluxo:

1. O usuário digita o nome do documento
2. Cola ou digita o conteúdo
3. Sistema salva na estrutura + realiza compressão

Exemplo:

Digite o nome do documento: doc1.txt
Digite o conteúdo do documento:
> A raposa rápida pula sobre o cão preguiçoso.

[✓] Documento armazenado com sucesso e comprimido!

3. Tela de Busca por Palavra-Chave

Componentes:

- Campo de busca
- Checkbox: ☐ Usar Suffix Array
- Lista de resultados com:
 - Nome do documento
 - Frequência
 - Trecho relevante

Exemplo:

Buscar palavra: "raposa"
[✓] Palavra encontrada em 2 documentos.

Resultados:
- doc1.txt (2 ocorrências)
 > "A raposa rápida pula..."
- doc3.txt (1 ocorrência)
 > "... onde a raposa espreita..."

4. Tela de Estatísticas e Ordenações

Componentes:

- Tabela de documentos:
 - Nome
 - Tamanho original
 - Tamanho comprimido
 - Data de criação

- Botões para ordenar por:
 - Nome
 - Tamanho
 - Data
- Gráficos simples de comparação (se GUI)

Exemplo:

Documentos ordenados por tamanho comprimido:

1. doc3.txt - 5.2KB
2. doc1.txt - 3.7KB
3. doc2.txt - 2.1KB

5. Tela de Gerenciamento de Estruturas

Componentes:

- Botões:
 - Atualizar índice invertido
 - Visualizar árvore AVL de palavras
 - Ver B-Tree de metadados
 - Forçar recompressão Huffman
- Árvore ou gráfico textual (para CLI)
- Logs de ações

Exemplo:

```
[AVL Tree]
  casa
  /  \
carro comida
  \
  cavalo
```

Índice invertido atualizado!

6. Tela de Detalhes de Documento

Componentes:

- Nome, data, tamanho original/comprimido
- Conteúdo do texto (descomprimido)
- Botões: [Excluir] [Recomprimir] [Visualizar índices]

Considerações

Essas telas podem ser implementadas com:

- * Java Swing / JavaFX para GUI
- * Lanterna / JLine / Text-UI para CLI com interface textual
- * Spring Shell ou apenas `Scanner + System.out` para CLI simples