Relatório EP AED I - Indexador e Buscador de Palavras

Nomes dos Integrantes:
Gustavo Henrique Ferreira Alves - NUSP 15674466
Guilherme Padun Santos - NUSP 13828007

1. Estrutura do código

O programa é estruturado da seguinte maneira:
indexador.c: O arquivo principal, onde a lógica de carregamento do arquivo, construção do índice e interação com o usuário é implementada.
Funções: O código usa funções para manipulação de palavras (remoção de pontuação, conversão para minúsculas), construção de árvores binárias e leitura do arquivo.
Arquivo de texto: O arquivo de texto é lido linha por linha, e cada palavra é indexada na árvore binária.

2. Exemplo de Saída

Após rodar o programa, o índice é construído e você pode realizar buscas para encontrar as palavras no arquivo. A saída pode ser algo como:

Arquivo: 'texto.txt'
Tipo de indice: 'arvore'
Numero de linhas no arquivo: 13
Total de palavras indexadas: XXXXX
Altura da arvore: XXXXX
Tempo de carga do arquivo e construcao do indice: XXXXX ms
> busca algorithm
Existem 4 ocorrências da palavra 'algorithm' na(s) seguinte(s) linha(s):
00001: Informally, an algorithm is any well-defined computational procedure that takes
00003: as output. An algorithm is thus a sequence of computational steps that transform
00006: We can also view an algorithm as a tool for solving a well-specified computational
00008: input/output relationship. The algorithm describes a specific computational
Tempo de busca: XXXXX ms
> busca set
Existem 2 ocorrências da palavra 'set' na(s) seguinte(s) linha(s):
00002: some value, or set of values, as input and produces some value, or set of values,
Tempo de busca: XXXXX ms
> busca quicksort
Palavra 'quicksort' nao encontrada.
Tempo de busca: XXXXX ms

> buska quicksort
Opcao invalida!
> fim

3. Testes

Nº palavras (aproximado)	Número de Linhas no Arquivo	Total de Palavras Indexadas	Estrutura	Altura da Árvore	Tempo de Carga do Índice (ms)	Tempo de Busca (ms)
100	8	75	Lista	-	0,49	0,07
			Árvore	7	0,93	0,11
1.000	50	528	Lista	-	2,42	0,31
			Árvore	11	3,56	0,32
10.000	1333	2640	Lista	-	27,07	0,99
			Árvore	14	13,63	1,01
110.000	15.604	13.634	Lista	-	73.491,00	107
			Árvore	17	133,00	91
750.000	32.369	28.299	Lista	-	100.908,57	0,49
			Árvore	18	831,59	0,14

Tempo de Busca: Árvores são mais eficientes em busca, especialmente em conjuntos maiores de dados.

Tempo de Construção do Índice: Árvores têm maior custo inicial para construção, mas compensam com buscas mais rápidas.

4. Análise dos Resultados

1. Tempo de Busca
    - Lista: O tempo de busca aumenta consideravelmente conforme o volume de dados cresce, refletindo a natureza linear das buscas em listas.
    - Árvore: O tempo de busca permanece relativamente estável, mesmo com conjuntos maiores de dados, devido à eficiência das operações de busca em árvores balanceadas (O(log n)).
2. Tempo de Construção do Índice

    - Lista: A construção do índice em listas é rápida, pois as palavras são adicionadas sequencialmente sem a necessidade de reordenação ou balanceamento.
    - Árvore: O tempo de construção é mais lento, já que envolve operações para manter a árvore balanceada, especialmente com o aumento do volume de dados.
3. Eficiência em Escalabilidade
    - A diferença de desempenho se torna mais evidente nos maiores conjuntos de dados (110.000 e 750.000 palavras). As árvores apresentam um tempo de busca substancialmente menor, enquanto o tempo de construção do índice ainda é aceitável considerando os ganhos em buscas rápidas.
4. Altura da Árvore
    - À medida que o número de palavras cresce, a altura da árvore aumenta, mas de forma controlada, graças às propriedades das árvores balanceadas. Isso garante que o desempenho permaneça eficiente.

Conclusão

- Para conjuntos menores de dados, as listas são uma escolha eficiente devido ao seu baixo custo de construção e simplicidade.
- Para conjuntos maiores, as árvores se destacam pela eficiência em buscas, tornando-se a melhor opção em cenários onde a performance de busca é crítica.

A escolha entre lista e árvore depende, portanto, do tamanho do conjunto de dados e da importância relativa entre tempo de construção e tempo de busca no sistema em questão.