

DISCENTES: Diego Machado de Moraes – 14652567 / Mariana Borges Araujo da Silva - 14596342

DOCENTE: Prof. Dr. Flávio Luiz Coutinho

TURMA: 94

DISCIPLINA: Análise e Estrutura de Dados I (ACH2023)

RELATÓRIO DO EXERCÍCIO-PROGRAMA

Neste exercício-programa foram implementadas duas formas de busca de palavras em arquivos de texto: busca por Lista Ligada e por Árvore Binária, ficando a critério do usuário selecionar qual forma de busca quer-se aplicar. Para apurar o tempo de execução das buscas em cada uma das formas (Lista Ligada ou Árvore Binária), foram realizados testes de execução com diversas entradas de variados tamanhos e a eficiência do funcionamento dos algoritmos foi apurada através da comparação do tempo de busca (em milissegundos) levado para exibir as informações sobre a palavra buscada nos textos.

O processo de criação de um índice baseado em uma Lista Ligada envolve operações de inserção, cuja complexidade assintótica é $O(n)$, que ocorre quando o elemento procurado não está na lista ou está na última posição. O melhor caso da complexidade assintótica de uma busca por Lista Ligada é $O(1)$, que ocorre quando a palavra buscada está na primeira posição da lista. Já o processo de criação de um índice baseado em uma Árvore Binária também envolve operações de inserção, cuja complexidade assintótica de tempo para busca, inserção e exclusão é de $O(\log n)$ no pior caso, onde n é o número de nós na árvore. A menor altura que uma árvore binária pode ter é $\Theta(\log(n))$, enquanto a maior altura seria $\Theta(n)$. A complexidade de tempo de busca no melhor caso é $O(1)$, quando o elemento procurado é a raiz da árvore. Sendo assim, a complexidade de tempo de busca é $O(h)$, onde h é a altura da árvore. Baseado nas análises assintóticas apresentadas, podemos concluir que a indexação/busca é mais eficiente quando feita em uma Árvore Binária. Essa conclusão foi feita a partir da observação do tempo de busca (em milissegundos) pela palavra no texto, como apresentado a seguir:

TESTE 1: text 1

- **Linhas:** 176;
- **Palavras:** 1541.
- **Lista Ligada:**
 - ➔ **Tempo para criação do índice:** 0139 ms;
 - ➔ **Tempo para busca de 'God':** 0003 ms;
 - ➔ **Tempo para busca de 'ashamed':** 0000 ms.
- **Árvore:**
 - ➔ **Tempo para criação do índice:** 0203 ms;
 - ➔ **Tempo para busca de 'God':** 0004 ms;
 - ➔ **Tempo para busca de 'ashamed':** 0000 ms.

TESTE 2: text 2

- **Linhas:** 711;
- **Palavras:** 10958.
- **Lista Ligada:**
 - ➔ **Tempo para criação do índice:** 0976 ms;
 - ➔ **Tempo para busca de 'formidable':** 0002 ms;
 - ➔ **Tempo para busca de 'sherlock':** 0015 ms.
- **Árvore:**
 - ➔ **Tempo para criação do índice:** 1254 ms;
 - ➔ **Tempo para busca de 'formidable':** 0000 ms;
 - ➔ **Tempo para busca de 'sherlock':** 0009 ms.

TESTE 3: text 3:

- **Linhas:** 992;
- **Palavras:** 16672.
- **Lista Ligada:**
 - ➔ **Tempo para criação do índice:** 1394 ms;
 - ➔ **Tempo para busca de 'formidable':** 0005 ms;
 - ➔ **Tempo para busca de 'sherlock':** 0020 ms.
- **Árvore:**
 - ➔ **Tempo para criação do índice:** 1360 ms;

- ➔ Tempo para busca de 'formidable': 0004 ms;
- ➔ Tempo para busca de 'sherlock': 0010 ms.

TESTE 4: text 4

- **Linhas:** 9784;
- **Palavras:** 85230.
- **Lista Ligada:**
 - ➔ Tempo para criação do índice: 9291 ms;
 - ➔ Tempo para busca de 'God': 0045 ms;
 - ➔ Tempo para busca de 'ashamed': 0000 ms.
- **Árvore:**
 - ➔ Tempo para criação do índice: 13322 ms;
 - ➔ Tempo para busca de 'God': 0048 ms;
 - ➔ Tempo para busca de 'ashamed': 0000 ms.

Diante dos testes realizados e das análises assintóticas, podemos concluir que, a Lista Ligada é mais eficiente na realização de testes para a criação de índices, o que contraria a lógica usual, já que a Árvore Binária é, na maioria dos casos, mais eficiente. Isso acontece por alguns motivos, dentre eles, podemos considerar que o cálculo da complexidade assintótica exclui as constantes. Já para a busca, a Árvore Binária é, geralmente, mais eficiente, mantendo o padrão esperado de uma busca na Árvore Binária.