****

TAMIRES BEATRIZ DA SILVA LUCENA (111866)

**ALGORITMOS E ESTRUTURAS DE DADOS II**

Profa. Lilian Berton

Recuperação de textos usando árvore B

SÃO JOSÉ DOS CAMPOS, SÃO PAULO.

2017

1. **Introdução**

Criado um repositório de textos com aproximadamente 30 arquivos no formato .txt, o objetivo do trabalho consiste em devolver ao usuário o(s) texto(s) que mais tenha relação (maior frequência de aparição) com uma certa palavra dada pelo usuário.

Os textos foram armazenados numa árvore B, para facilitar as buscas.

1. **Metodologia**

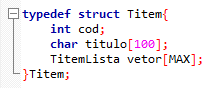
**2.1. Materiais**

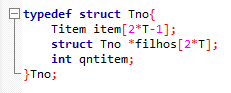
Foi utilizada a estrutura de dados árvore B com T = 2 que respeita os seguintes critérios:

* Cada nó possui no máximo 2\*T-1 itens (3 itens);
* Cada nó possui no máximo 2\*T filhos (4 filhos);
* Cada item possui valor maior que os itens de sua direita e menor que os itens de sua direita;
* Os nós possuem todos os valores de seus itens maiores que de seu filho da esquerda e menores que de seu filho da direita;
* Os nós folhas estão todos no mesmo nível da árvore.

Armazenamento:

Realizada a leitura do texto, foi gerada uma cópia que não contém as stopwords e pontuações.

Foi criado um vetor com todas as palavras e suas respectivas frequências, e armazenado tal vetor numa struct contendo também o título do texto e seu código:  


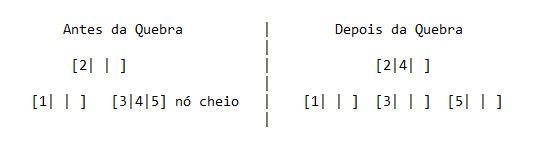


Este item, por sua vez, está armazenado em uma struct denominada Tno, que administra a quantidade de itens em cada nó e seus filhos.

**2.2. Métodos**

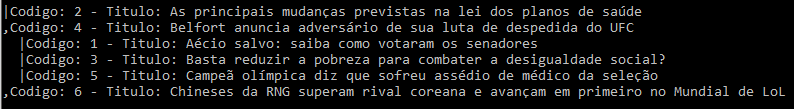
- Inserção

Para inserir um novo item, a árvore é percorrida a procura do nó ideal. Caso o nó onde o novo item se encaixe esteja cheio (2\*T-1 itens), é necessário que seja feita a quebra do nó em questão, subindo o item do meio para o nó-pai desse nó que foi quebrado.  
Caso o nó a ser “quebrado” seja raiz, a árvore aumenta em altura.

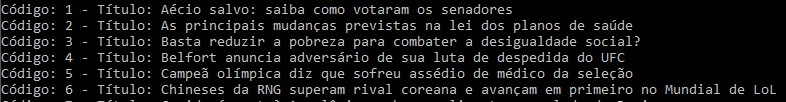


- Impressão

*Pré-Ordem:*

O método de Impressão, percorre todos os nós e imprime separando por vírgulas os itens que estão no mesmo nó e os espações são proporcionais à diferença de nível entre o nó e seus filhos. 

*Em Ordem:*

Percorre todos os nós imprimindo-os em ordem.  


- Remoção

O método remoção, percorre a árvore em busca do item escolhido, e tal remoção deve ser realizada sempre em um nó-folha.

Quando o item que deseja remover não está em um nó-folha, deve ser realizada a troca de lugares e aí então, quando o item desejado estiver em um nó-folha, realizar a remoção.

*Casos:*

1. Se o item a ser removido estiver em um nó-folha e este mantiver o número mínimo de itens após a remoção, basta removê-lo.

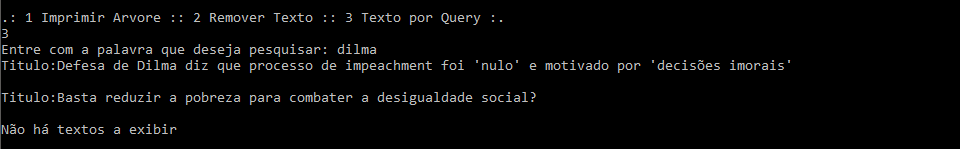
2. Se o item não estiver em uma folha, devemos substituí-lo pelo seu antecessor, ou seja, o item do nó-folha mais à direita da árvore-filho à esquerda.

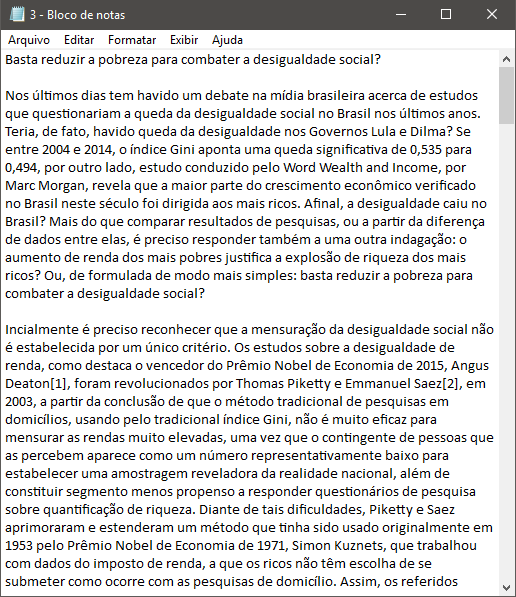
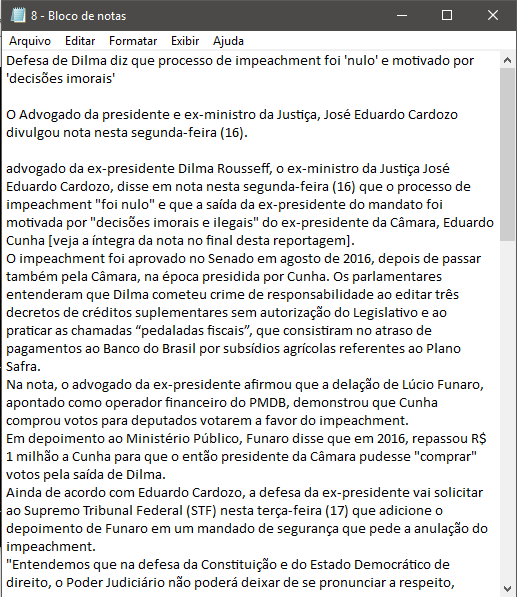
3. Se a folha a qual foi retirada o item de substituição, não ficar com o mínimo de ocupação, um nó-irmão deve ceder a essa um de seus itens (passa o menor item do irmão esquerdo para o nó-pai e o menor do nó-pai para o nó desfalcado ou como segunda opção, o inverso: o irmão direito passa o maior para o nó-pai e o maior do nó-pai para o nó desfalcado).

4. Caso os irmãos não possam ceder algum item, deve ser realizada a união do nó que possui o item a ser removido com algum de seus irmãos (para isso, passamos algum item do meio do nó-pai para o nó-remoção e trazemos os itens que estão no nó-irmão escolhido para o nó-remoção também).

Y6

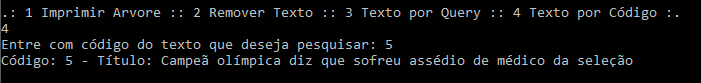
- Pesquisa Query

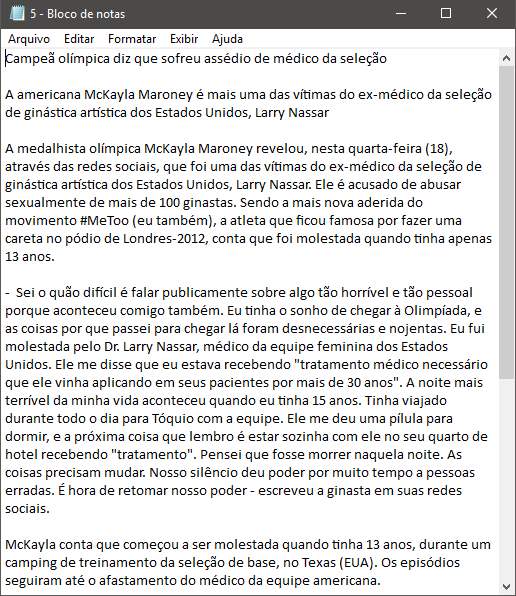
Dada uma palavra de entrada fornecida pelo usuário, o método de pesquisa percorre todos os itens de todos os nós da árvore b a procura de itens que possuam a palavra pesquisada com frequência maior do que 2 (definida no algoritmo como uma frequência relevante) e exibe o(s) texto(s) encontrado(s) na tela.



- Pesquisa Código

É possível ainda, pesquisar um texto apenas fornecendo seu código. O método percorre a árvore de maneira otimizada, passando pelo caminho mais curto possível até o item desejado, visto que os itens da árvore estão devidamente organizados/inclusos em ordem crescente.





1. **Conclusão**

Neste trabalho pudemos perceber a complexidade da estrutura árvore B, e entender como e porque ela funciona bem, mantendo uma boa altura que permite buscas mais eficientes.  
Além disso foi possível explorar outras técnicas de programação, como, por exemplo, manuseio de arquivos.

Finalmente, a estrutura viabilizou a query (consulta) e exibição dos textos em tela de maneira efetiva atendendo às expectativas.

1. **Referências**

MARCOS ANDRÉ SILVEIRA KUTOVA. **Remoção em uma Árvore B.** Disponível em vídeo na plataforma YouTube: <https://www.youtube.com/watch?v=OhTbpo1WLes>

BERTON, L. Slide de Aula. Aula 14 – Árvores B