

O objetivo deste trabalho consiste em contar a frequência das palavras de um texto usando árvores binárias de busca (ABB) e árvores binárias de busca balanceadas (AVL). Adicionalmente, outras informações relacionadas também são solicitadas.

As palavras encontram-se em arquivos do tipo texto, que devem ser considerados como entrada para seu código. Os arquivos devem ser utilizados da forma fornecida, sem nenhum tipo de pré-processamento ou alteração em seu conteúdo.

A contagem de frequência e demais informações devem ser realizadas de forma individual para cada arquivo. Os arquivos fornecidos possuem os seguintes nomes (nomes reduzidos):

- **mam-na-np-lo.txt (mam)**
- **exame_10000-na-np-lo.txt (exame)**
- **biblia-na-np-lo.txt (bíblia)**

Para atingir o objetivo deste trabalho, você deve escrever seu programa em C++, utilizando classes de ABB e AVL. Seu código pode incluir os métodos fornecidos em aula assim como pode incorporar seu próprio código de forma a obter todas as informações solicitadas, que devem ser exibidas como *output* do seu programa por meio do método **write()**. O pseudo-código para isso é dado a seguir:

1. ler o arquivo de entrada
2. criar uma ABB/AVL com as palavras contidas no texto (sem repetições)
3. escrever os resultados para preencher a tabela abaixo (método **write()**)

No passo 2 do pseudo-código, as palavras devem ser inseridas na ABB/AVL à medida que aparecem no texto de entrada, utilizando o algoritmo de *busca com inserção* (método **searchInsert()**) em sua implementação recursiva.

Por exemplo, o código básico em C++ fornecido a seguir efetua o processamento para o primeiro dos arquivos fornecidos:

```
#include <iostream>
#include <fstream>
#include <cstring>
using namespace std;

int main()
{ BinarySearchTree<string> bst;
  AVLTree<string> avl;
  string word;
  ifstream theInput("mam-na-np-lo.txt");

  // ler cada palavra do arquivo e inserir na ABB e na AVL
  while( theInput >> word )
  { bst.searchInsert(word);
    avl.searchInsert(word);
  }
  theInput.close();
  bst.write();
  avl.write();

  return 0;
}
```

Caso você prefira escrever o resultado para um arquivo de saída em vez da saída padrão (tela), pode utilizar o seguinte fragmento de código nos métodos **write()**:

```
ofstream theOutput("mam-na-np-lo.output.txt");
theOutput << "Escrever as informações solicitadas..." << endl;
```

```
theOutput.close();
```

Ao final da inserção de todas as palavras no respectivo texto na ABB e AVL, os métodos **write()** devem escrever as informações solicitadas a seguir. As informações são exibidas no formato de tabela para facilitar o entendimento do que está sendo solicitado. Todavia, nenhuma das tabelas fornecidas deve ser preenchida por você. *As respostas devem ser colocadas diretamente no Exercício correspondente no TIDIA-AE.*

Submeta, no TIDIA-AE, seu código sob a forma de arquivo compactado .ZIP contendo todo o código-fonte escrito para obter as informações solicitadas neste trabalho. Informe a forma de execução, em arquivo README.TXT, para obter os resultados para cada arquivo fornecido para análise.

Com base no código em C++ implementado para este trabalho, forneça as seguintes métricas a respeito de ABB e AVL.

Preencha a tabela seguinte com os resultados obtidos:

		mam	exame	bíblia
A	Palavras distintas			
B	Palavras			
C	Altura da árvore mínima			
D	Altura ABB			
E	Folhas ABB			
F	Comparações de palavras na busca com inserção ABB			
G	Altura AVL			
H	Folhas AVL			
I	Comparações de palavras na busca com inserção AVL			
J	Rotações na busca com inserção AVL			

Preencha a tabela seguinte com os resultados da tabela anterior. Represente números com duas casas decimais.

		mam	exame	Bíblia
	Média de comparações por palavra ABB = F/B			
	Média de comparações por palavra AVL = I/B			
	Média de rotações por palavra distinta AVL = J/A			

Indique, a partir do seu código, as 10 palavras mais frequentes (em ordem decrescente de frequência), bem como suas respectivas frequências. Em caso de palavras com a mesma frequência, coloque-as em ordem alfabética no preenchimento de cada tabela.

a)

mam

	Palavra	Frequência
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

b)

exame

	Palavra	Frequência
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

c)

biblia

	Palavra	Frequência
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		