

O objetivo deste laboratório consiste em contar a frequência das palavras de um texto usando árvores binárias de busca (ABB).

Para realizar o que está sendo solicitado, o programa de realizar os seguintes passos:

1. ler o arquivo de entrada
2. criar uma ABB com as palavras contidas no texto (sem repetições)
3. escrever cada palavra e sua respectiva frequência no texto, em ordem alfabética

O passo 1 pode ser executado utilizando-se o seguinte exemplo:

```
int main()
{ BinarySearchTree t;
  string s;
  ifstream ent("mam.txt");
  ofstream saida("saida-mam.txt");

  while(ent >> s)
  { t.SearchInsert(s);
  }
  ent.close();

  saida << "Escrever as informações solicitadas..." << endl;
  t.Print();
  saida.close();
  return 0;
}
```

No passo 2, as palavras devem ser inseridas na ABB à medida que aparecem no texto de entrada, utilizando o algoritmo de *busca com inserção* (método *SearchInsert*), implementado na forma *iterativa*. Indique também os seguintes resultados na tabela seguinte, após a inserção de todas as palavras no texto na ABB:

	mam.txt	exame_10000.txt	genesis.txt
Altura			
Altura da árvore mínima			
Número total de palavras			
Número total de palavras distintas			
Número de folhas			
Número total de comparações			

O total de comparações consiste no número de comparações de palavras que foram efetuadas para realizar a busca com inserção na ABB. Ao terminar, o programa desenvolvido e em execução deve ser mostrado ao professor, juntamente com as saídas necessárias para preencher a tabela anterior.

Compare e comente os valores obtidos para a primeira e segunda linha da tabela em cada situação, ou seja, o tamanho da árvore obtida foi próximo ao da árvore ideal? Explique.

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.