

**Título:**

Análise Experimental e Comparativa de Árvores Binárias de Busca: BST Clássica, AVL e Rubro-Negra

**Contextualização**

As estruturas de árvores binárias de busca são fundamentais para a organização e recuperação eficiente de dados em Ciência da Computação. Entre as variações mais conhecidas estão a árvore binária de busca clássica (BST), a árvore AVL e a árvore Rubro-Negra, cada uma com características distintas de balanceamento e desempenho. Este trabalho propõe um estudo experimental dessas estruturas, utilizando dados reais, para que você compreenda, na prática, as diferenças de eficiência e comportamento dessas árvores em diferentes cenários.

**Objetivos**

- Implementar em linguagem C as estruturas de BST clássica, AVL e Rubro-Negra, com suporte às operações de inserção e busca.
- Instrumentar as implementações para medir e registrar o tempo de execução das operações, o número de rotações (quando aplicável) e a altura final das árvores.
- Utilizar conjuntos de dados reais, obtidos do repositório de Princeton, para conduzir experimentos de inserção e busca em diferentes cenários (dados ordenados e não ordenados).
- Analisar e comparar os resultados obtidos, relacionando-os com a teoria estudada em sala de aula.

**Descrição das Tarefas**

- 1 Implementação das Estruturas
  - 1.1 Implemente as três estruturas (BST, AVL e Rubro-Negra) em arquivos separados, seguindo boas práticas de programação modular em C.
  - 1.2 Cada estrutura deve conter funções para:
    - Inserção de elementos (strings)
    - Busca de elementos (strings)
    - Cálculo da altura da árvore
  - 1.3 No caso das árvores AVL e Rubro-Negra, contabilize o número de rotações realizadas durante as inserções
- 2 Preparação dos Dados
  - Baixe o arquivo words.utf-8.txt do site <https://introcs.cs.princeton.edu/java/data/words.utf-8.txt>
  - Gere uma versão ordenada do arquivo;
- 3 Execução dos Experimentos
  - 3.1 Para cada estrutura e para cada arquivo de dados:
    - Meça e registre o tempo total de inserção de todos os elementos do arquivo na árvore.
    - Meça e registre o tempo total para realizar 100 buscas aleatórias de palavras presentes no arquivo.
    - Meça e registre a altura final da árvore após todas as inserções.
    - Meça e registre o número total de rotações realizadas durante as inserções (apenas para AVL e Rubro-Negra).
- 4 Relatório
  - 4.1 Apresente os resultados em tabelas e gráficos comparativos.
  - 4.2 Analise criticamente os resultados, discutindo:
    - O impacto do balanceamento automático nas operações de inserção e busca.
    - As diferenças observadas entre os cenários de dados ordenados e não ordenados.
    - A relação entre o número de rotações e o desempenho das árvores balanceadas.

- Como a altura da árvore influencia o tempo de busca.

4.3 Baseado nos dados, responda às seguintes perguntas:

- Por que a BST clássica apresenta desempenho inferior em dados ordenados?
- Por que a AVL tende a realizar mais rotações que a Rubro-Negra?
- Em quais cenários a Rubro-Negra pode ser preferida à AVL?

## 5 Entrega

5.1 Código-fonte bem documentado das três estruturas.

5.2 Scripts ou instruções para compilação e execução dos experimentos.

5.3 Relatório técnico em PDF, contendo:

- Introdução teórica breve sobre cada estrutura
- Metodologia experimental
- Resultados (tabelas e gráficos)
- Discussão e respostas às questões propostas

## Conclusão

### Critérios de Avaliação

- Corretude e eficiência das implementações (30%)
- Instrumentação e coleta adequada das métricas (20%)
- Qualidade da análise dos resultados e das respostas às questões (30%)
- Clareza, organização e apresentação do relatório (15%)
- Documentação e facilidade de execução do código (5%)

### Observações

O trabalho deve ser realizado em duplas.

Utilize apenas bibliotecas padrão da linguagem C.