

## TRABALHO PRÁTICO PARTE 3

**PRAZO PARA ENTREGA: 18/08/2021**

### 1. Introdução

O objetivo principal deste trabalho é dar ao aluno a oportunidade de analisar e comparar o desempenho de diferentes estruturas de dados aplicadas a um conjunto de dados reais. Espera-se que o aluno seja capaz de:

- Manipular adequadamente arquivos em modo texto e binário;
- Compreender e implementar corretamente diferentes árvores de busca;
- Analisar o desempenho das estruturas implementadas segundo o referencial teórico adquirido na disciplina;
- Apresentar os resultados obtidos com o trabalho de maneira formal.

### 2. Desenvolvimento

As etapas para o desenvolvimento do trabalho são descritas abaixo.

#### ***Etapas 1: Carregamento dos dados***

Nesta etapa deve utilizar a implementação feita nas entregas anteriores para recuperar registros aleatórios do arquivo binário. Neste trabalho, usaremos apenas a base de dados de artistas.

#### ***Etapas 2: Implementação de estruturas de dados balanceadas***

Você deverá implementar duas estruturas de dados em árvore:

- Árvore Vermelho-Preto
- Árvore B.

O tipo abstrato de dados deve implementar as operações de inserção e busca (não há necessidade de implementar a remoção). Cada nó deve armazenar o código do artista e a sua localização no arquivo binário, mas a ordenação da árvore será determinada pelo (nome do artista). As outras informações do registro, quando necessárias, serão recuperadas através do acesso ao disco.

### ***Etapa 3: Análise das estruturas balanceadas***

Nesta etapa, você irá comparar o desempenho da operação de busca nas estruturas implementadas. Para cada estrutura, o **seu programa deverá indexar todos os registros** de artistas na estrutura de dados. Após as inserções, você deverá realizar a busca indicada abaixo.

- Buscar um artista pelo nome na base de dados e imprimir todas as informações do registro.

Durante as buscas, deverão ser computados o total de comparações entre chaves realizadas. Além disso, os tempos de execução das inserções e das buscas deverão ser medidos. Para gerar as estatísticas de desempenho, você deverá executar a busca acima para M diferentes registros aleatórios extraídos do arquivo binário. Minimamente, utilize 100. Ao final, compute as médias de cada uma das métricas (comparações e tempo). Salve todos os resultados obtidos em um arquivo `saida.txt`, contendo tanto os resultados individuais quanto a média final.

As estruturas de dados que devem ser implementadas são as seguintes:

- Árvore Vermelho-Preto
- Árvore B ( $t = 20$ )
- Árvore B ( $t = 200$ )

Considere que a ordem  $t$  da árvore B representa o número **mínimo** de chaves em cada nó (exceto a raiz), conforme visto em sala de aula. Não há necessidade de se manter preso aos valores de  $t$  especificados acima, o grupo pode adotar valores diferentes (ou mais que dois valores) para os testes. Tenha o cuidado, no entanto, de escolher valores que permitam avaliar a diferença no desempenho para árvores de ordem baixa e árvores de ordem alta.

### ***Etapa 4 - Programa principal***

O programa deve oferecer ao usuário um menu para permitir a escolha de qual estrutura de dados balanceada será executada:

1. Árvore Vermelho-Preta
2. Árvore B
3. Sair

Se a opção 2 for a escolhida, então o usuário deve indicar a ordem mínima da árvore B que será usada. Para ambas opções, o programa deve pedir ao usuário a escolha entre o modo de análise, que gera o relatório da etapa 3 ou modo de teste, a qual realiza a busca de um artista cujo nome deve ser lido pelo teclado.

### 3. Relatório

Você deverá confeccionar um relatório detalhado sobre o trabalho desenvolvido. Este relatório deve conter, obrigatoriamente, os seguintes itens:

- Detalhamento das atividades realizadas por cada membro do grupo;
- Explicações dos algoritmos escolhidos, bem como as razões para cada escolha;
- Explicações das estruturas de dados implementadas, bem como as razões para as decisões de implementação;
- Apresentação dos resultados através de tabelas e gráficos que permitam visualizar com clareza o que foi obtido;
- Análise detalhada dos resultados obtidos;
- Toda e qualquer referência utilizada no desenvolvimento do trabalho.

Note que o relatório deve ser formal, bem organizado e bem redigido. A divisão de tarefas do trabalho se aplica somente à implementação. Todo o grupo é responsável pelo relatório (escrita e revisão).

### 4. Exigências

O trabalho deverá, obrigatoriamente, atender aos seguintes requisitos:

- Implementação em C ou C++
- O projeto deve ser compilável e executável via linha de comando. Não conte com a presença de IDEs como Code::Blocks, Visual Studio ou NetBeans. Caso seu grupo opte por utilizar algum ambiente de desenvolvimento, certifique-se de que o projeto enviado possa também ser facilmente compilado em um sistema operacional Linux sem esses ambientes instalados. Forneça instruções claras e precisas de compilação e execução pela linha de comando. **Recomenda-se a utilização de algum Makefile ou script para a compilação.** Caso o grupo julgue necessário, é possível solicitar que o professor verifique as instruções de compilação **antes** do prazo final de envio.
- **Trabalho entregue após o prazo será penalizado.**
- **O programa desenvolvido deve permitir que o usuário entre com o caminho do diretório que contém os arquivos de dados como um argumento na linha de comando.** Veja o Exemplo:

```
$ ./programa /diretorio/contendo/arquivos/entrada
```

- Assim, como nos trabalhos anteriores, o programa deve procurar pela existência dos arquivos binários dentro da pasta. Se não existir, deve pré-processá-los (etapa 0) antes de seguir para as etapas seguintes. Não é permitida a utilização de bibliotecas externas. As estruturas de dados e seus algoritmos devem ser implementados pelo

grupo. Um dos objetivos do trabalho é que vocês aprendam a trabalhar com os algoritmos escolhidos. Se você tiver dúvida quanto à utilização de alguma função ou biblioteca, entre em contato com o professor.

- Obviamente, todo código deve ser de autoria do grupo. Não é permitida a utilização de códigos de terceiros ou de outros grupos. É permitida a pesquisa por estratégias para a solução dos problemas (e as referências utilizadas nessas pesquisas devem constar do relatório), porém a apropriação de código alheio não será aceita. **Qualquer tentativa de plágio identificada resultará em nota zero. Os códigos fontes serão analisados pelo sistema Moss (<http://theory.stanford.edu/~aiken/moss/>)**

## 5. Entrega

O grupo deverá ser formado por **no máximo** 4 alunos, e as responsabilidades de cada aluno devem ser documentadas e registradas. Não é permitido que algum integrante do grupo fique responsável somente pela confecção do relatório. Todos os integrantes devem contribuir com a implementação. A distribuição das responsabilidades deve ser feita de maneira uniforme, de modo que cada membro do grupo se envolva com o trabalho na mesma proporção que os demais.

Todos os itens abaixo devem ser entregues:

1. Código-fonte completo;
  - a. Deve ser submetido como um tag release 3.0 no repositório git (github) do grupo;
  - b. **Não incluir o dataset na submissão via Google Classroom (tanto o .csv quando .bin).**
2. Relatório em Google Doc ou PDF atendendo ao especificado na Seção 3 deste documento.

## 6. Critérios de avaliação

O grupo será avaliado de acordo com os seguintes critérios:

- Execução correta do código (E);
- Atendimento ao que foi solicitado (A) (valor entre 0 e 1);
- Organização do código (O): seu código deve estar bem modularizado e bem documentado;
- Qualidade do relatório apresentado (R).

Cada membro do grupo será avaliado individualmente, tanto com relação aos detalhes de implementação que ficaram sob sua responsabilidade, quanto ao entendimento em alto nível de abstração do que foi feito pelo grupo como um todo. O entendimento teórico do

conteúdo relacionado ao trabalho também será avaliado. A nota individual (M) será um valor de 0 a 1 que irá ponderar a nota da implementação.

A nota final de cada integrante será computada de acordo com a seguinte fórmula:

$$\text{Nota} = 0.5 * [(0.7 * E + 0.3 * O) * M * A] + 0.5 * R * A$$

O critério (A) será considerado de acordo com a composição do grupo. O objetivo é evitar que algum aluno seja prejudicado pela desistência de outros membros do grupo. **Caso algum membro do seu grupo tranque ou abandone a disciplina, comunique o professor o quanto antes para que se possa discutir alternativas.**