

Trabalho 3º Bimestre: Algoritmos de Árvores

Projeto e Análise de Algoritmos - PAA

2023

André Luiz Brun¹

¹Colegiado de Ciência da Computação
Campus de Cascavel - UNIOESTE

Resumo. *Este documento consiste na especificação formal do terceiro trabalho da disciplina de Projeto e Análise de Algoritmos (Csc2152) para o ano letivo de 2023. Nesta avaliação serão aplicados conceitos acerca das estratégias de tratamento de estruturas hierárquicas, as árvores. Neste contexto serão exploradas estratégias sem balanceamento, com balanceamento a partir de fatores de coloração (RB) e com base no fator de balanceamento (AVL). Aqui são apresentadas as atividades a serem desenvolvidas e como cada processo deverá ser realizado. Além disso, o documento contém as informações sobre a formação das equipes, o objeto de trabalho de cada uma e as datas de entrega e apresentação dos relatórios.*

O objetivo do terceiro trabalho da disciplina consiste em comparar o comportamento, em termos práticos, de diferentes estratégias de implementação para tratar dados de forma hierárquica. Busca-se determinar o **Custo Empírico** de cada implementação identificando-se o **tempo gasto** e **número de comparações** entre abordagens concorrentes. Neste experimento busca-se evidenciar quando cada estratégia é mais adequada frente à(s) sua(s) concorrente(s) através de experimentos práticos.

1. Guilherme Marchiotti Ferraz e Roberval Requiao Junior

1.1. Objetivo

Aplicar e corroborar conceitos adquiridos sobre algoritmos de manipulação de dados de forma hierárquica. Para tanto, deverão ser implementadas duas soluções para árvores binárias de busca.

Na primeira solução deverá ser implementada uma **Árvore Binária de Busca (ABB)**, sem balanceamento.

Em seguida, afim de comparação, deverá ser implementada a estratégia de **Rubro Negra (RB)**.

1.2. Critérios de avaliação

Para realizar a comparação dos métodos a análise deverá ser feita em duas etapas:

1. Na primeira, construção, deverá ser estimada a quantidade de comparações entre chaves para descobrir a posição em que o elemento deve ser inserido e o tempo gasto para construir a estrutura inteira;
2. Na segunda, consulta, deverá ser contabilizado o número de comparações para se buscar os elementos na estrutura construída na etapa anterior;
3. Além da contagem de comparações deverão ser contabilizados os tempos gastos para se construir cada uma das árvores, bem como os tempos necessários para se consultar todos os elementos presentes no arquivo de entrada.

1.3. Como

- As linguagens que podem ser utilizadas no desenvolvimento do trabalho são: C, C++, Java e Python;
- A mesma linguagem deve ser adotada para ambos os métodos. Além disso, deve-se empregar as mesmas estratégias para a realização do trabalho;
- A forma com que os métodos serão implementados é determinada pelo grupo;
- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis no link a seguir: Link Grupo 1;

2. Gabriel Norato Claro, Gabriel Alves Mazzuco e Rodrigo Brickmann Rocha

2.1. Objetivo

Aplicar e corroborar conceitos adquiridos sobre algoritmos de manipulação de dados de forma hierárquica. Para tanto, deverão ser implementadas duas soluções para árvores binárias de busca.

Inicialmente deverá ser construída uma árvore de busca binária **AVL** que servirá de estrutura para as buscas a serem avaliadas, ou seja, os desempenhos observados estarão atrelados ao comportamento da árvore obtida.

Na primeira solução deverá ser implementada uma **Busca em Profundidade (DFS)**. Em seguida, afim de comparação, deverá ser implementada a estratégia de **Busca em Largura (BFS)**. A ideia é comparar as duas abordagens em termos cronológicos e em número de comparações entre chaves para se realizar operações de busca.

2.2. Critérios de avaliação

Para realizar a comparação dos métodos a análise deverá ser feita em três etapas:

1. Na primeira, construção, deverá ser construída uma **Árvore AVL** a qual servirá de base para as duas opções de busca;
2. Na segunda, consulta DFS, deverá ser contabilizado o número de comparações para se buscar os elementos na estrutura construída na etapa anterior. Além disso, deverá ser contabilizado o tempo cronológico gasto para buscar todos os elementos do arquivo de consulta.
3. Na terceira etapa o processo será similar à etapa 2, porém, neste momento será implementada e avaliada uma busca em largura;

2.3. Como

- As linguagens que podem ser utilizadas no desenvolvimento do trabalho são: C, C++, Java e Python;
- A mesma linguagem deve ser adotada para ambos os métodos. Além disso, deve-se empregar as mesmas estratégias para a realização do trabalho;
- A forma com que os métodos serão implementados é determinada pelo grupo;
- Considerem como vértice inicial da busca sempre aquele de menor índice;
- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis no link a seguir: [Link Grupo 2](#);

3. Vinicius Drage, Gustavo Martini e Gustavo Macedo

3.1. Objetivo

Aplicar e corroborar conceitos adquiridos sobre algoritmos de manipulação de dados de forma hierárquica. Para tanto, deverão ser implementadas duas soluções para árvores binárias de busca.

Na primeira solução deverá ser implementada uma **Árvore Binária de Busca (ABB)**, sem balanceamento.

Em seguida, afim de comparação, deverá ser implementada a estratégia de **Árvore AVL**.

3.2. Critérios de avaliação

Para realizar a comparação dos métodos a análise deverá ser feita em duas etapas:

1. Na primeira, construção, deverá ser estimada a quantidade de comparações entre chaves para descobrir a posição em que o elemento deve ser inserido e o tempo gasto para construir a estrutura inteira;
2. Na segunda, consulta, deverá ser contabilizado o número de comparações para se buscar os elementos na estrutura construída na etapa anterior;
3. Além da contagem de comparações deverão ser contabilizados os tempos gastos para se construir cada uma das árvores, bem como os tempos necessários para se consultar todos os elementos presentes no arquivo de entrada.

3.3. Como

- As linguagens que podem ser utilizadas no desenvolvimento do trabalho são: C, C++, Java e Python;
- A mesma linguagem deve ser adotada para ambos os métodos. Além disso, deve-se empregar as mesmas estratégias para a realização do trabalho;
- A forma com que os métodos serão implementados é determinada pelo grupo;
- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis no link a seguir: Link Grupo 3;

4. Gustavo Henrique Marchezoni Orlandini, Lucas David Tomalack de Souza e Rafael Roberto Hoffmann

4.1. Objetivo

Aplicar e corroborar conceitos adquiridos sobre algoritmos de manipulação de dados de forma hierárquica. Para tanto, deverão ser implementadas duas soluções para árvores binárias de busca.

Inicialmente deverá ser construída uma árvore de busca binária **Rubro Negra (RB)** que servirá de estrutura para as buscas a serem avaliadas, ou seja, os desempenhos observados estarão atrelados ao comportamento da árvore obtida.

Na primeira solução deverá ser implementada uma **Busca em Profundidade (DFS)**. Em seguida, afim de comparação, deverá ser implementada a estratégia de **Busca em Largura (BFS)**. A ideia é comparar as duas abordagens em termos cronológicos e em número de comparações entre chaves para se realizar operações de busca.

4.2. Critérios de avaliação

Para realizar a comparação dos métodos a análise deverá ser feita em três etapas:

1. Na primeira, construção, deverá ser construída uma **Rubro Negra** a qual servirá de base para as duas opções de busca;
2. Na segunda, consulta DFS, deverá ser contabilizado o número de comparações para se buscar os elementos na estrutura construída na etapa anterior. Além disso, deverá ser contabilizado o tempo cronológico gasto para buscar todos os elementos do arquivo de consulta.
3. Na terceira etapa o processo será similar à etapa 2, porém, neste momento será implementada e avaliada uma busca em largura;

4.3. Como

- As linguagens que podem ser utilizadas no desenvolvimento do trabalho são: C, C++, Java e Python;
- A mesma linguagem deve ser adotada para ambos os métodos. Além disso, deve-se empregar as mesmas estratégias para a realização do trabalho;
- A forma com que os métodos serão implementados é determinada pelo grupo;
- Considerem como vértice inicial da busca sempre aquele de menor índice;
- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis no link a seguir: [Link Grupo 4](#);

5. Jaqueline Cavaller Faino, Davi Marchetti Giacomel e Maria Eduarda Crema Carlos

5.1. Objetivo

Aplicar e corroborar conceitos adquiridos sobre algoritmos de manipulação de dados de forma hierárquica. Para tanto, deverão ser implementadas duas soluções para árvores binárias de busca.

Na primeira solução deverá ser implementada uma **Árvore Binária de Busca (ABB)** utilizando uma abordagem recursiva.

Em seguida, afim de comparação, deverá ser implementado o mesmo tipo de árvore adotando, no entanto, uma estratégia iterativa.

5.2. Critérios de avaliação

Para realizar a comparação dos métodos a análise deverá ser feita em duas etapas:

1. Na primeira, construção, deverá ser estimada a quantidade de comparações entre chaves para descobrir a posição em que o elemento deve ser inserido e o tempo gasto para construir a estrutura inteira;
2. Na segunda, consulta, deverá ser contabilizado o número de comparações para se buscar os elementos na estrutura construída na etapa anterior;
3. Além da contagem de comparações deverão ser contabilizados os tempos gastos para se construir cada uma das árvores, bem como os tempos necessários para se consultar todos os elementos presentes no arquivo de entrada.

5.3. Como

- As linguagens que podem ser utilizadas no desenvolvimento do trabalho são: C, C++, Java e Python;
- A mesma linguagem deve ser adotada para ambos os métodos. Além disso, deve-se empregar as mesmas estratégias para a realização do trabalho;
- A forma com que os métodos serão implementados é determinada pelo grupo;
- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis no link a seguir: [Link Grupo 5](#);

6. Heloisa Aparecida Alves, Eduarda Elger e Ellen Carine Bonafin Marques

6.1. Objetivo

Aplicar e corroborar conceitos adquiridos sobre algoritmos de manipulação de dados de forma hierárquica. Para tanto, deverão ser implementadas duas soluções para árvores binárias de busca.

Na primeira solução deverá ser implementada uma **Árvore AVL**. Em seguida, afim de comparação, deverá ser implementada a estratégia de **Árvore Rubro Negra (RB)**.

6.2. Critérios de avaliação

Para realizar a comparação dos métodos a análise deverá ser feita em duas etapas:

1. Na primeira, **construção**, deverá ser estimada a quantidade de comparações entre chaves para descobrir a posição em que o elemento deve ser inserido e o tempo gasto para construir a estrutura inteira;
2. Na segunda, **consulta**, deverá ser contabilizado o número de comparações e o tempo gasto para se buscar os elementos na estrutura construída na etapa anterior;
3. Além da contagem de comparações deverão ser contabilizados os tempos gastos para se construir cada uma das árvores, bem como os tempos necessários para se consultar todos os elementos presentes no arquivo de entrada.

6.3. Como

- As linguagens que podem ser utilizadas no desenvolvimento do trabalho são: C, C++, Java e Python;
- A mesma linguagem deve ser adotada para ambos os métodos. Além disso, deve-se empregar as mesmas estratégias para a realização do trabalho;
- A forma com que os métodos serão implementados é determinada pelo grupo;
- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis no link a seguir: Link Grupo 6;

7. Gabriel Santos Ramos, Lucas Becker da Conceição e Rafael Gotz

7.1. Objetivo

Aplicar e corroborar conceitos adquiridos sobre algoritmos de manipulação de dados de forma hierárquica. Para tanto, deverão ser implementadas duas soluções para árvores binárias de busca.

Inicialmente deverá ser construída uma **árvore de busca binária sem balanceamento** que servirá de estrutura para as buscas a serem avaliadas, ou seja, os desempenhos observados estarão atrelados ao comportamento da árvore obtida.

Na primeira solução deverá ser implementada uma **Busca em Profundidade (DFS)**. Em seguida, afim de comparação, deverá ser implementada a estratégia de **Busca em Largura (BFS)**. A ideia é comparar as duas abordagens em termos cronológicos e em número de comparações entre chaves para se realizar operações de busca.

7.2. Critérios de avaliação

Para realizar a comparação dos métodos a análise deverá ser feita em três etapas:

1. Na primeira, construção, deverá ser construída uma **Árvore de Busca binária sem balanceamento** a qual servirá de base para as duas opções de busca;
2. Na segunda, consulta DFS, deverá ser contabilizado o número de comparações para se buscar os elementos na estrutura construída na etapa anterior. Além disso, deverá ser contabilizado o tempo cronológico gasto para buscar todos os elementos do arquivo de consulta.
3. Na terceira etapa o processo será similar à etapa 2, porém, neste momento será implementada e avaliada uma busca em largura;

7.3. Como

- As linguagens que podem ser utilizadas no desenvolvimento do trabalho são: C, C++, Java e Python;
- A mesma linguagem deve ser adotada para ambos os métodos. Além disso, deve-se empregar as mesmas estratégias para a realização do trabalho;
- A forma com que os métodos serão implementados é determinada pelo grupo;
- Considerem como vértice inicial da busca sempre aquele de menor índice;
- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis no link a seguir: [Link Grupo 7](#);

8. Alonso Lucca Fritz e Lucca Abbado Neres

8.1. Objetivo

Aplicar e corroborar conceitos adquiridos sobre algoritmos de manipulação de dados de forma hierárquica. Para tanto, deverão ser implementadas duas soluções para árvores binárias de busca.

Inicialmente deverá ser construída uma **árvore de busca binária sem balanceamento** que servirá de estrutura para as buscas a serem avaliadas, ou seja, os desempenhos observados estarão atrelados ao comportamento da árvore obtida.

Na primeira solução deverá ser implementada uma **Busca Recursiva**. Em seguida, afim de comparação, deverá ser implementada a estratégia de **Busca Iterativa**. A ideia é comparar as duas abordagens em termos cronológicos e em número de comparações entre chaves para se realizar operações de busca.

8.2. Critérios de avaliação

Para realizar a comparação dos métodos a análise deverá ser feita em três etapas:

1. Na primeira, construção, deverá ser construída uma **Árvore de Busca binária sem balanceamento** a qual servirá de base para as duas opções de busca;
2. Na segunda, consulta executada recursivamente, deverá ser contabilizado o número de comparações para se buscar os elementos na estrutura construída na etapa anterior. Além disso, deverá ser contabilizado o tempo cronológico gasto para buscar todos os elementos do arquivo de consulta.
3. Na terceira etapa o processo será similar à etapa 2, porém, neste momento será implementada e avaliada uma estratégia de consulta iterativa;

8.3. Como

- As linguagens que podem ser utilizadas no desenvolvimento do trabalho são: C, C++, Java e Python;
- A mesma linguagem deve ser adotada para ambos os métodos. Além disso, deve-se empregar as mesmas estratégias para a realização do trabalho;
- A forma com que os métodos serão implementados é determinada pelo grupo;
- Considerem como vértice inicial da busca sempre aquele de menor índice;
- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis no link a seguir: Link Grupo 8;

9. Arthur Augusto Eich e Gustavo Pauli da Luz

9.1. Objetivo

Aplicar e corroborar conceitos adquiridos sobre algoritmos de manipulação de dados de forma hierárquica. Para tanto, deverão ser implementadas duas soluções para árvores binárias de busca.

Na primeira solução deverá ser implementada uma **Árvore Binária de Busca (ABB)**, sem balanceamento.

Em seguida, afim de comparação, deverá ser implementada a estratégia de **Rubro Negra (RB)**.

9.2. Critérios de avaliação

Para realizar a comparação dos métodos a análise deverá ser feita em duas etapas:

1. Na primeira, construção, deverá ser estimada a quantidade de comparações entre chaves para descobrir a posição em que o elemento deve ser inserido e o tempo gasto para construir a estrutura inteira;
2. Na segunda, consulta, deverá ser contabilizado o número de comparações para se buscar os elementos na estrutura construída na etapa anterior;
3. Além da contagem de comparações deverão ser contabilizados os tempos gastos para se construir cada uma das árvores, bem como os tempos necessários para se consultar todos os elementos presentes no arquivo de entrada.

9.3. Como

- As linguagens que podem ser utilizadas no desenvolvimento do trabalho são: C, C++, Java e Python;
- A mesma linguagem deve ser adotada para ambos os métodos. Além disso, deve-se empregar as mesmas estratégias para a realização do trabalho;
- A forma com que os métodos serão implementados é determinada pelo grupo;
- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis no link a seguir: Link Grupo 9;

10. Relatório

Deve ser elaborado um relatório técnico em formato pdf contendo:

- A Descrição de como foi realizado o processo empírico de determinação dos custos: cenário de realização dos experimentos e como foram tomadas as métricas exigidas;
 - Detalhar a configuração usada nos testes (Processador, SO, IDE, etc.);
- Gráficos evidenciando o comportamento dos métodos perante todos os cenários considerando o tamanho dos conjuntos de entrada;
- **Devem ser construídos gráficos que explicitem os comportamentos em termos de tempo gasto na execução dos métodos, bem como gráficos que evidenciem o número de comparações entre chaves;**
- o Análise do comportamento dos métodos durante a execução dos testes;
 - Esta análise deve ser feita com bastante critério e ser esclarecedora, apontando razões para os comportamentos observados;
 - **AS figuras e gráficos devem ser invocadas no texto e explicadas;**

O formato do relatório deve ser a formatação presente neste texto. As regras para tal podem ser obtidas no link download. No arquivo disponível pode-se utilizar a formatação em arquivo .doc ou em latex.

11. Código-fonte

Além do relatório citado, cada equipe deverá enviar os códigos fonte construídos para a execução dos experimentos. Ambos arquivos podem ser compactados e enviados como arquivo único.

Não é necessário o envio do projeto compilado, apenas do código fonte base.

Neste código devem constar os comandos utilizados para tomadas de tempo de execução e contagem das variáveis de interesse.

12. Para quando?

O trabalho deverá ser submetido no link disponibilizado na turma de disciplina dentro do ambiente Microsoft Teams até as **23:59 do dia 02/06/2024**.