

Estrutura de Dados e Análise de Algoritmos

2025

André Luiz Brun¹

¹Programa de Pós-graduação em Ciência da Computação (PPGComp)
Campus de Cascavel - UNIOESTE

Resumo. *Este documento consiste na especificação formal do trabalho da disciplina de Estrutura de Dados e Análise de Algoritmos para o ano letivo de 2025/1. Nesta avaliação serão aplicados conceitos acerca das estratégias de ordenação de conjuntos de dados, tratamento de estruturas hierárquicas e algoritmos baseados em grafos. Neste contexto serão exploradas estratégias de busca em conjuntos de dados, abordagens hierárquicas como árvores sem balanceamento e com balanceamento a partir de fatores de coloração (RB). Além disso, serão explorados métodos de ordenação com custo assintótico $O(n^2)$ e $O(n \log_2 n)$, bem como algoritmos clássicos de manipulação de grafos. 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 deste trabalho consiste em comparar o comportamento, em termos práticos, de diferentes estratégias de implementação para tratar os contextos citados acima. Busca-se determinar o **Custo Empírico** de cada implementação identificando-se o **tempo cronológico gasto** e **número de comparações entre chaves** (quando exigido) 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.

Na seções de 1 a 7 são apresentadas as especificações do tema de cada equipe e, na Seção 8 é detalhado como deverá ser construído o relatório e envio do trabalho.

1. José Lucas Errera Gomes
Rafaela Louise dos Santos
Vivian Miwa Fugihara

1.1. Objetivo

Aplicar e corroborar conceitos adquiridos sobre algoritmos de busca em conjuntos de dados de forma a, em caso positivo, identificar a posição em que o elemento desejado se encontra e, em caso negativo, informar que o elemento buscado não está presente no conjunto de dados.

Para tanto deverão ser implementadas três diferentes estratégias de busca:

- Busca linear com sentinela;
- Busca binária;
- Jump Search.

Os algoritmos receberão um conjunto de entrada (A) que servirá de base para as operações de consulta e, em seguida, um segundo conjunto de dados (B) contento as chaves a serem buscadas no primeiro conjunto.

Para a **Busca Linear com Sentinela** não há necessidade de ordenação do conjunto A. Por outro lado, a **Busca Binária** e **Jump Search** necessitam que o conjunto buscado esteja ordenado de forma crescente. Como esta é uma exigência destes métodos, o tempo gasto na ordenação inicial deve ser considerado.

1.2. Critérios de avaliação

Para realizar a comparação dos métodos a análise deverá ser feita sob duas óticas. Em um primeiro momento serão comparados o tempo (cronológico) gasto para que os algoritmos sejam executados sobre todos os vértices dos grafos. Em seguida deverão ser comparados o número de comparações necessárias para a busca de todos os elementos presentes no conjunto B.

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 todos os métodos. Além disso, deve-se empregar as mesmas estratégias para a realização do trabalho;
- Os três algoritmos devem ser executados na mesma máquina, sob as mesmas condições;
- 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. Bruno Felipe Maihach
Davi Marchetti Giacomel
Jaqueline Cavaller Faino

2.1. Objetivo

Aplicar e corroborar conceitos adquiridos sobre algoritmos de roteamento cujo objetivo é encontrar o menor caminho entre todos os pares de vértices de um grafo. Para tanto, deverão ser implementadas duas estratégias para obtenção dos menores caminhos.

Na primeira solução deverá ser implementado o algoritmo de **Floyd-Warshall**. Em seguida, afim de comparação, deverá ser implementado o método de **Dijkstra** que será executado V vezes. Em cada execução do algoritmo um novo vértice é usado como ponto de partida. Dessa forma, será possível encontrar o menor caminho entre todos os pares de vértices.

2.2. Critérios de avaliação

Para realizar a comparação dos métodos a análise deverá ser feita sob duas óticas. Em um primeiro momento serão comparados o tempo (cronológico) gasto para que os algoritmos sejam executados sobre todos os vértices dos grafos. Em seguida deverão ser comparadas as distâncias obtidas pelos métodos implementados. Espera-se que ambos cheguem ao mesmo custo.

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;
- Os dois algoritmos devem ser executados na mesma máquina;
- 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 2](#);

3. Angela Maria de Souza
Cleonice Maria Pereira
Erico de Araujo

3.1. Objetivo

Aplicar e corroborar conceitos adquiridos com relação aos métodos de ordenação de ordem quadrática, bem como ilustrar o comportamento apresentados pelas estratégias em diferentes cenários.

A tarefa consiste em executar os algoritmos de ordenação **BubbleSort**, **SelectionSort** e **InsertionSort** sobre diversos conjuntos de dados.

3.2. Critérios de avaliação

Deve-se avaliar o comportamento dos métodos perante conjuntos de testes com diferentes características. Durante a execução dos experimentos o critério de análise será o **tempo cronológico** gasto para a execução dos métodos.

Cada algoritmo deverá ser executado sobre 4 conjuntos de dados distintos:

- Aleatório: os elementos presentes nos conjuntos não seguem padrões determinados;
- Ordenado: os elementos já estão distribuídos de forma crescente no conjunto;
- Parcialmente ordenado: o comportamento dos elementos é um misto entre aleatórios e ordenados;
- Decrescente: os elementos estão distribuídos de forma decrescente.

Assim, a análise cronológica a ser feita deve apresentar os comportamentos:

- BubbleSort sobre os quatro tipos de entrada;
- SelectionSort sobre os quatro tipos de entrada;
- InsertionSort sobre os quatro tipos de entrada;
- Análise do desempenho dos três métodos sobre o conjunto de dados ordenado;
- Análise do desempenho dos três métodos sobre o conjunto de dados parcialmente ordenado;
- Análise do desempenho dos três métodos sobre o conjunto de dados aleatórios;
- Análise do desempenho dos três métodos sobre o conjunto de dados decrescentes.

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 nos links a seguir Aleatórios; Ordenados; Parcialmente Ordenados; Decrescentes;

4. Edinéia dos Santos Brizola Brum

Jefferson Rodrigo Speck

Rafael Ferreira Lima

4.1. Objetivo

Aplicar e corroborar conceitos adquiridos com relação aos métodos de ordenação de ordem $O(n \log_2 n)$, bem como ilustrar o comportamento apresentados pelas estratégias em diferentes cenários.

A tarefa consiste em executar os algoritmos de ordenação **QuickSort**, **MergeSort** e **HeapSort** sobre diversos conjuntos de dados.

4.2. Critérios de avaliação

Deve-se avaliar o comportamento dos métodos perante conjuntos de testes com diferentes características. Durante a execução dos experimentos o critério de análise será o **tempo cronológico** gasto para a execução dos métodos.

Cada algoritmo deverá ser executado sobre 4 conjuntos de dados distintos:

- Aleatório: os elementos presentes nos conjuntos não seguem padrões determinados;
- Ordenado: os elementos já estão distribuídos de forma crescente no conjunto;
- Parcialmente ordenado: o comportamento dos elementos é um misto entre aleatórios e ordenados;
- Decrescente: os elementos estão distribuídos de forma decrescente.

Assim, a análise cronológica a ser feita deve apresentar os comportamentos:

- QuickSort sobre os quatro tipos de entrada;
- MergeSort sobre os quatro tipos de entrada;
- HeapSort sobre os quatro tipos de entrada;
- Análise do desempenho dos três métodos sobre o conjunto de dados ordenado;
- Análise do desempenho dos três métodos sobre o conjunto de dados parcialmente ordenado;
- Análise do desempenho dos três métodos sobre o conjunto de dados aleatórios;
- Análise do desempenho dos três métodos sobre o conjunto de dados decrescentes.

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;
- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis nos links a seguir Aleatórios; Ordenados; Parcialmente Ordenados; Decrescentes;

5. Luciano Jeyson Santos da Rocha

Tiago Luis Custódio

Lucca Abbado Neres

5.1. Objetivo

Aplicar e corroborar conceitos adquiridos sobre algoritmos de manipulação de dados armazenados na forma de grafos. Para tanto, deverão ser implementadas duas estratégias de busca em grafos:

- Implementar o algoritmo de Busca em Profundidade (*Depth-First Search* - DFS) sobre um grafo não direcionado;
- Implementar o algoritmo de Busca em Largura (*Breadth-First Search* - BFS) para percorrer um grafo não direcionado.

5.2. Critérios de avaliação

- Tempo cronológico para a execução do algoritmo de Busca em Largura;
- Tempo cronológico para a execução do algoritmo de Busca em Profundidade;
- Tamanho máximo (em número de elementos e não em espaço de memória) alcançado pela pilha ou fila ao longo da execução de cada busca.

5.3. Como

- Os arquivos de entrada contém a matriz de adjacência de um grafo ponderado não direcionado;
- 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 5](#);

<p>6. Luiz Fernando Becher de Araujo Roberval Requião Júnior Marcos Agnes</p>
--

6.1. Objetivo

Aplicar e corroborar conceitos adquiridos sobre algoritmos de roteamento cujo objetivo é encontrar o menor caminho a partir de uma origem até todos os demais vértices de um grafo. Para tanto, deverão ser implementadas duas estratégias para obtenção dos menores caminhos.

Na primeira solução deverá ser implementado o algoritmo de **Dijkstra**. Em seguida, afim de comparação, deverá ser implementada o método de **Bellman-Ford**.

6.2. Critérios de avaliação

Para realizar a comparação dos métodos a análise deverá ser feita sob duas óticas. Em um primeiro momento serão comparados o tempo (cronológico) gasto para que os algoritmos sejam executados sobre todos os vértices dos grafos. Em seguida deverão ser comparadas as distâncias obtidas pelos métodos implementados. Espera-se que ambos cheguem ao mesmo custo.

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;
- Considerem como vértice inicial da busca sempre aquele de menor índice;
- Os dois algoritmos devem ser executados na mesma máquina;
- 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](#);

<p>7. Felipe Talini Leandro Ferreira Bueno Marcos Antonio Guido</p>
--

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.

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)**.

7.2. 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.

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;
- Todos os experimentos devem ser executados na mesma máquina sob as mesmas condições;
- A entrada dos dados deve ser feita com base nos arquivos texto disponíveis no link a seguir: [Link Grupo 7](#);

8. 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 da arquitetura usada nos testes (Processador, SO, IDE, etc..);
- **Deverão ser executadas 7 repetições dos experimentos, sendo descartadas as de melhor e pior tempo para cada método. A análise então será realizada sobre a média das cinco execuções remanescentes;**
- 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 analisados durante os experimentos**
- o Análise do comportamento dos métodos durante a execução dos testes;
 - Esta análise deve ser feita com bastante critério e ser esclarecedora, apon-tando razões para os comportamentos observados;
 - **As figuras e gráficos devem ser invocados no texto e explicados;**

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

Não há limites mínimo ou máximo para o número de páginas.

9. 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.

10. Para quando?

O trabalho deverá ser submetido no link disponibilizado na turma de disciplina dentro do ambiente Microsoft Teams até as **12:00 do dia 25/07/2025**.

As apresentações ocorrerão então no dia **28/07/2024 a partir das 13:30** e dia 29/07/2024 (caso necessário) a partir das 08:00. Tais encontros ocorrerão no Laboratório 7 do bloco D (mesmo que temos utilizado nas aulas).

Uma vez que é desejado que todas as equipes apresentem seu trabalho na aula do dia 28/07/2025, o tempo limite de apresentação para grupo será de 15 minutos.