



Projeto e Análise de Algoritmos
Trabalho 3º Bimestre
Árvores

O trabalho está dividido em várias propostas. Cada grupo, através de sorteio, ficou responsável por resolver uma proposta das listadas a seguir.

O QUE DEVE SER ENTREGUE

Relatório em formato pdf contendo:

- 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.
- 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.
- O formato dos relatórios deve seguir o padrão da SBC conforme disponível no endereço: <https://www.sbc.org.br/documentos-da-sbc/summary/169-templates-para-artigos-e-capitulos-de-livros/878-modelosparapublicaodeartigos>



Códigos fonte usados na execução dos experimentos

- Não é necessário o envio dos projetos, apenas os códigos fonte;
- Manter no código os comandos usados para tomada de tempo e contagem do número de comparações.

Quando:

A data de entrega está marcada para dia 12/07/2022 até as 23:59.

A entrega do trabalho deve ser feita via Teams.

No dia 18/07/2022 haverá então a apresentação das soluções desenvolvidas. A ordem com que as apresentações serão realizadas é dada pela sequência 7, 4, 9, 8, 6, 3, 10, 2, 1, 5. Caso no horário de início da apresentação a equipe não estiver presente, o grupo perde o direito de fazer sua apresentação.

A qualidade da apresentação também constitui critério de avaliação.

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação aos métodos de balanceamento de árvores e seu impacto no custo de sua manipulação em termos cronológicos e número de comparações.

Tarefa:

Implementar as seguintes estratégias de construção de árvores binárias:

- a) Árvore Binária de Busca (ABB);
- b) Árvore AVL.

Avaliar o comportamento dos métodos perante um conjunto de testes com diferentes características.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pelo grupo. No entanto, deve-se adotar a mesma abordagem para ambos os métodos.
- A entrada dos dados para a **construção** das árvores deve ser feita com base nos arquivos texto disponíveis no link a seguir (arquivos do tipo txt com nome no formato ConstruirN.txt).
https://drive.google.com/file/d/1Nww_7s6a4Q6k6XtNqJ0O4fMp5UpOMifB/view?usp=sharing
- Os valores a serem utilizados no momento da **consulta** estão disponíveis no mesmo link (arquivos do tipo txt com formato ConsultarN.txt).
- Critérios que devem ser analisados:
 - a) Durante a Construção das árvores;
 - Tempo cronológico gasto na construção das estruturas
 - Número total de comparações entre chaves realizadas para construir cada uma das estruturas (não é necessário levar em conta o teste que verifica se o ponteiro é null).
 - b) Durante a Consulta às árvores
 - Tempo cronológico gasto na consulta de **todos os elementos** presentes no arquivo dentro das duas estruturas construídas anteriormente
 - Número de comparações realizadas entre chaves para consultar **todos os elementos** dos arquivos de consulta nas árvores formadas anteriormente (não é necessário levar em conta o teste que verifica se o ponteiro é null).

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação aos métodos de balanceamento de árvores e seu impacto no custo de sua manipulação em termos de tempo e número de comparações.

Tarefa:

Implementar as seguintes estratégias de construção de árvores binárias:

- a) Árvore AVL;
- b) Árvore Rubro Negra.

Avaliar o comportamento dos métodos perante um conjunto de testes com diferentes características.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pelo grupo. No entanto, deve-se adotar a mesma abordagem para ambos os métodos.
- A entrada dos dados para a **construção** das árvores deve ser feita com base nos arquivos texto disponíveis no link a seguir (arquivos do tipo txt com nome no formato ConstruirN.txt).
<https://drive.google.com/file/d/1AKfCDajrZPLbXzMQAH3S1TN6fLEhRnJ/view?usp=sharing>
- Os valores a serem utilizados no momento da **consulta** estão disponíveis no mesmo link (arquivos do tipo txt com formato ConsultarN.txt).
- Critérios que devem ser analisados:
 - a) Durante a Construção das árvores;
 - Tempo cronológico gasto na construção das estruturas
 - Número total de comparações entre chaves realizadas para construir cada uma das estruturas (não é necessário levar em conta o teste que verifica se o ponteiro é null).
 - b) Durante a Consulta às árvores
 - Tempo cronológico gasto na consulta de **todos os elementos** presentes no arquivo dentro das duas estruturas construídas anteriormente
 - Número de comparações realizadas entre chaves para consultar **todos os elementos** dos arquivos de consulta nas árvores formadas anteriormente (não é necessário levar em conta o teste que verifica se o ponteiro é null).

PROPOSTA 3 - Igor Engler e Marcos Mucelini

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação à manipulação de diferentes estruturas de dados. Neste caso a aplicação visa comparar o comportamento de uma lista dinâmica ordenada frente à uma Árvore AVL.

Tarefa:

Implementar as seguintes estratégias de armazenamento de dados:

- a) Árvore AVL;
- b) Lista dinâmica ordenada.

Avaliar o comportamento dos métodos perante um conjunto de testes com diferentes características.

A lista deve se manter ordenada no momento da construção de forma que cada novo elemento seja inserido exatamente na posição correta.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pelo grupo. No entanto, deve-se adotar a mesma abordagem para ambos os métodos.
- Caso seja de interesse do grupo, após construída a lista contendo todos os elementos, a mesma pode ser convertida em um vetor estático a fim de realizar as operações de consulta usando uma busca binária. Em caso contrário a busca a ser realizada será a linear (not fair!).
- A entrada dos dados para a **construção** das árvores deve ser feita com base nos arquivos texto disponíveis no link a seguir (arquivos do tipo txt com nome no formato ConstruirN.txt).
https://drive.google.com/file/d/1NFALXJuAwp_x61n8sovS5qLUpgySPF/view?usp=sharing
- Os valores a serem utilizados no momento da **consulta** estão disponíveis no mesmo link (arquivos do tipo txt com formato ConsultarN.txt).
- Critérios que devem ser analisados:
 - a) Durante a Construção das estruturas
 - Tempo cronológico gasto na construção das estruturas
 - Número de comparações entre chaves realizadas para construir cada uma das estruturas (não é necessário levar em conta o teste que verifica se o ponteiro é null).
 - b) Durante a etapa de Consulta
 - Tempo cronológico gasto na consulta de **todos os elementos** presentes no arquivo dentro das duas estruturas construídas anteriormente
 - Número de comparações realizadas entre chaves para consultar **todos os elementos** dos arquivos de consulta nas estruturas formadas anteriormente (não é necessário levar em conta o teste que verifica se o ponteiro é null).

PROPOSTA 4 – João Dota, Rodrigo Campos e Valquíria Belusso

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação à manipulação de diferentes estruturas de dados. Neste caso a aplicação visa comparar o comportamento de uma lista dinâmica ordenada frente à uma Árvore Rubro Negra.

Tarefa:

Implementar as seguintes estratégias de armazenamento de dados:

- a) Árvore Rubro Negra;
- b) Lista dinâmica ordenada.

Avaliar o comportamento dos métodos perante um conjunto de testes com diferentes características.

A lista deve se manter ordenada no momento da construção de forma que cada novo elemento seja inserido exatamente na posição correta.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pelo grupo.
- Caso seja de interesse do grupo, após construída a lista contendo todos os elementos, a mesma pode ser convertida em um vetor estático a fim de realizar as operações de consulta usando uma busca binária. Em caso contrário a busca a ser realizada será a linear.
- A entrada dos dados para a **construção** das árvores deve ser feita com base nos arquivos texto disponíveis no link a seguir (arquivos do tipo txt com nome no formato ConstruirN.txt).
<https://drive.google.com/file/d/10Zqfa63fdcg3VjkuMNCUfKGdH3pK4Zrq/view?usp=sharing>
- Os valores a serem utilizados no momento da **consulta** estão disponíveis no mesmo link (arquivos do tipo txt com formato ConsultarN.txt).
- Critérios que devem ser analisados:
 - a) Durante a Construção das estruturas
 - Tempo cronológico gasto na construção das estruturas
 - Número de comparações entre chaves realizadas para construir cada uma das estruturas (não é necessário levar em conta o teste que verifica se o ponteiro é null).
 - b) Durante a etapa de Consulta
 - Tempo cronológico gasto na consulta de **todos os elementos** presentes no arquivo dentro das duas estruturas construídas anteriormente
 - Número de comparações realizadas entre chaves para consultar **todos os elementos** dos arquivos de consulta nas estruturas formadas anteriormente (não é necessário levar em conta o teste que verifica se o ponteiro é null).

PROPOSTA 5 – Leonardo Vanzin, Lucas Veit e Roberta Aparecida Alcantara

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação à manipulação de diferentes estruturas de dados. Neste caso a aplicação visa comparar o comportamento dos métodos de busca em largura (amplitude) e profundidade sobre uma Árvore de Busca Binária sem balanceamento e uma Árvore AVL.

Tarefa:

Implementar as seguintes estratégias de busca em árvores:

- a) Busca em Profundidade (DFS);
- b) Busca em Largura ou Amplitude (BFS).

Implementar as seguintes estratégias de construção de árvores binárias:

- a) Árvore AVL;
- b) Árvore de Busca Binária.

Avaliar o comportamento dos métodos perante um conjunto de testes com diferentes características.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pelo grupo.
- A entrada dos dados para a **construção** das árvores deve ser feita com base nos arquivos texto disponíveis no link a seguir (arquivos do tipo txt com nome no formato ConstruirN.txt).
<https://drive.google.com/file/d/12w3acP01QJbJwGPiXhBkYXN2j3YOnUBT/view?usp=sharing>
- Os valores a serem utilizados no momento das **buscas** estão disponíveis no mesmo link (arquivos do tipo txt com formato ConsultarN.txt).
- Critérios que devem ser analisados:
 - Tempo cronológico gasto na busca em profundidade para as duas estruturas construídas;
 - Número de comparações (ou chaves processadas) na busca em profundidade dentro das duas estruturas (não é necessário levar em conta o teste que verifica se o ponteiro é null).
 - Tempo cronológico gasto na busca em largura para as duas estruturas construídas;
 - Número de comparações (ou chaves processadas) na busca em largura dentro das duas estruturas (não é necessário levar em conta o teste que verifica se o ponteiro é null).

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação aos métodos de balanceamento de árvores e seu impacto no custo de sua manipulação em termos cronológicos e número de comparações.

Tarefa:

Implementar as seguintes estratégias de construção de árvores binárias:

- a) Árvore AVL;
- b) Árvore Rubro Negra.

Avaliar o comportamento dos métodos perante um conjunto de testes com diferentes características.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pelo grupo. No entanto, deve-se adotar a mesma abordagem para ambos os métodos.
- A entrada dos dados para a **construção** das árvores deve ser feita com base nos arquivos texto disponíveis no link a seguir (arquivos do tipo txt com nome no formato ConstruirN.txt).
https://drive.google.com/file/d/14qaBGmk_XC8CYEcJdH-MszZXGQ8yBo6B/view?usp=sharing
- Os valores a serem utilizados no momento da **consulta** estão disponíveis no mesmo link (arquivos do tipo txt com formato ConsultarN.txt).
- Critérios que devem ser analisados:
 - a) Durante a Construção das árvores;
 - Tempo cronológico gasto na construção das estruturas
 - Número total de comparações entre chaves realizadas para construir cada uma das estruturas (não é necessário levar em conta o teste que verifica se o ponteiro é null).
 - b) Durante a Consulta às árvores
 - Tempo cronológico gasto na consulta de **todos os elementos** presentes no arquivo dentro das duas estruturas construídas anteriormente
 - Número de comparações realizadas entre chaves para consultar **todos os elementos** dos arquivos de consulta nas árvores formadas anteriormente (não é necessário levar em conta o teste que verifica se o ponteiro é null).

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação à manipulação e implementação de diferentes estruturas de dados. Neste caso a aplicação visa comparar o comportamento dos métodos de construção de uma árvore de busca binária (sem balanceamento) de forma recursiva perante a construção da mesma estrutura porém de forma iterativa.

Tarefa:

Implementar as seguintes estratégias de construção de árvores binárias:

- a) Árvore Binária de Busca recursiva;
- b) Árvore Binária de Busca iterativa;

Implementar as seguintes estratégias de consulta em árvores binárias:

- a) Busca recursiva;
- b) Busca iterativa;

Avaliar o comportamento dos métodos perante um conjunto de testes com diferentes características.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pelo grupo. No entanto, deve-se adotar a mesma abordagem para ambos os métodos.
- A entrada dos dados para a **construção** das árvores deve ser feita com base nos arquivos texto disponíveis no link a seguir (arquivos do tipo txt com nome no formato ConstruirN.txt).
https://drive.google.com/file/d/1C5XQk9h7Mf5_somRxCXBRwATV6i58ldj/view?usp=sharing
- Os valores a serem utilizados no momento da **consulta** estão disponíveis no mesmo link (arquivos do tipo txt com formato ConsultarN.txt).
- Critérios que devem ser analisados:
 - a) Durante a Construção das árvores;
 - Tempo cronológico gasto na construção das estruturas
 - Número de comparações entre chaves realizadas para construir cada uma das estruturas.
 - b) Durante a Consulta às árvores
Depois de construídas, as árvores terão a mesma estrutura, independente de terem sido construídas recursivamente ou de forma iterativa. Assim, a consulta será realizada apenas na árvore construída iterativamente (ao longo dos experimentos vocês descobrirão a razão). Serão comparadas as duas opções de busca sobre a mesma estrutura construída.
 - Tempo cronológico gasto na consulta de **todos os elementos** presentes no arquivo dentro das duas estruturas construídas anteriormente
 - Número de comparações realizadas entre chaves para consultar **todos os elementos** dos arquivos de consulta nas árvores formadas anteriormente

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação à manipulação de elementos em memória. Neste caso, analisar o comportamento do método de ordenação HeapSort.

Tarefa:

Implementar as seguintes estratégias de ordenação:

- a) HeapSort baseado em um Heap de máximo;
- b) HeapSort baseado em um Heap de mínimo.

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

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pelo grupo. No entanto, deve-se adotar a mesma abordagem para ambos os métodos.

Devem ser construídos quatro conjuntos de testes, disponíveis no link abaixo:
<https://drive.google.com/file/d/1DX2MJ7WBjzLbOEEA93cR5TYjJMzT8cU3/view?usp=sharing>

- Os dois métodos de ordenação trabalhando sobre conjunto aleatórios;
- Os dois métodos de ordenação trabalhando sobre conjuntos ordenados de forma decrescente;
- Todos os métodos sobre conjuntos ordenados de forma crescente;
- Todos os métodos sobre conjuntos parcialmente ordenados de forma crescente.

Além disso, devem ser analisado o comportamento de cada técnica sobre diferentes entradas conforme descrito a seguir:

- Max-HeapSort sobre os quatro tipos de entrada;
- Min-HeapSort sobre os quatro tipos de entrada;

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação à manipulação de diferentes estruturas de dados. Neste caso a aplicação visa comparar o comportamento dos métodos liberação de memória aplicados às estruturas de dados de Árvore de Busca Binária AVL frente às árvores Rubro Negras.

Tarefa:

Implementar as seguintes estratégias de construção de árvores binárias de busca:

- a) Árvores AVL;
- b) Árvores Rubro Negras (RN).

Implementar um mecanismo de liberação de memória que é responsável por liberar o espaço ocupado por cada um dos elementos presentes nas estruturas construídas (AVL e RN).

Avaliar o comportamento dos métodos perante um conjunto de testes com diferentes características.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A forma com que os métodos serão implementados é determinada pelo grupo. No entanto, deve-se adotar a mesma abordagem para ambos os métodos.
- A entrada dos dados para a **construção** das árvores deve ser feita com base nos arquivos texto disponíveis no link a seguir (arquivos do tipo txt com nome no formato ConstruirN.txt).
https://drive.google.com/file/d/12XslgGaGrXfA4e1CrLOBgb4c-fCknJ_U/view?usp=sharing
- Critérios que devem ser analisados:
 - Tempo cronológico gasto na liberação de todos os elementos presentes nas estruturas (AVL e RN);
 - Número de comparações gastas para encontrar e liberar todos os elementos nas duas estruturas (AVL e RN).

Objetivo:

Aplicar e corroborar conceitos adquiridos com relação à manipulação de diferentes estruturas de dados. Neste caso a aplicação visa comparar o comportamento de uma função de Hashing implementada usando tratamento de colisões usando encadeamento e usando endereçamento aberto.

Tarefa:

Implementar as seguintes estratégias de tratamento de colisões para uma função hash:

- a) Tratamento usando encadeamento;
- b) Tratamento usando endereçamento aberto.

Avaliar o comportamento dos métodos perante um conjunto de testes com diferentes características.

Como:

- A linguagem utilizada no desenvolvimento é de vossa escolha.
- A função de Hash proposta é de escolha da equipe. Para tanto, considerem que os dados de entrada são apenas números do tipo inteiro.
- A forma com que os métodos serão implementados é determinada pelo grupo. No entanto, deve-se adotar a mesma abordagem para ambos os métodos.
- A entrada dos dados para a **construção** das tabelas hash deve ser feita com base nos arquivos texto disponíveis no link a seguir (arquivos do tipo txt com nome no formato ConstruirN.txt).
https://drive.google.com/file/d/1VCA_FmXD1N-NYzz5lpjmebVK4Lm1PT3-/view?usp=sharing
- Os valores a serem utilizados no momento da **consulta** estão disponíveis no mesmo link (arquivos do tipo txt com formato ConsultarN.txt).
- Critérios que devem ser analisados:
 - a) Durante a Construção das estruturas
 - Tempo cronológico gasto na construção das duas estruturas
 - Número de comparações entre chaves realizadas para construir cada uma das estruturas (não é necessário levar em conta o teste que verifica se o ponteiro é null).
 - b) Durante a etapa de Consulta
 - Tempo cronológico gasto na consulta de **todos os elementos** presentes no arquivo dentro das duas estruturas construídas anteriormente
 - Número de comparações realizadas entre chaves para consultar **todos os elementos** dos arquivos de consulta nas estruturas formadas anteriormente (não é necessário levar em conta o teste que verifica se o ponteiro é null).