



**UNIVERSIDADE DE SÃO PAULO**  
**INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO**  
**INTRODUÇÃO À CIÊNCIA DE COMPUTAÇÃO II**

**LEANDRO SATOSHI DE SIQUEIRA**  
**GABRIEL SILVA FONTES**

**ÁRVORE DE BUSCA BINÁRIA**

**SÃO CARLOS - SÃO PAULO**

**2018**

LEANDRO SATOSHI DE SIQUEIRA  
GABRIEL SILVA FONTES

## ÁRVORE DE BUSCA BINÁRIA

Trabalho apresentado ao Curso de Bacharelado em Sistema de Informação do Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo.

Orientador: Adenilso da Silva Simão

SÃO CARLOS - SÃO PAULO

2018

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>3</b>
<b>2</b>	<b>METODOLOGIA</b>	<b>4</b>
<b>3</b>	<b>A ESTRUTURA</b>	<b>5</b>
3.1	DEFINIÇÃO	5
3.2	ELEMENTOS E TERMINOLOGIAS	5
<b>4</b>	<b>DISCUSSÕES E RESULTADOS</b>	<b>6</b>
	<b>REFERÊNCIAS</b>	<b>8</b>

## **1 INTRODUÇÃO**

Este trabalho refere-se à implementação da árvore de busca binária e suas funções estudadas nas aulas da disciplina de Introdução à ciência de computação II.

Primeiramente será feita uma análise de suas funcionalidades e gastos de tempo e memória para os mesmos, e em seguida algumas discussões em relação a implementação e resultados obtidos.

## **2 METODOLOGIA**

A metodologia utilizada foi um estudo prévio da estrutura de dados e suas funcionalidades, seguido de sua implementação e de testes práticos para validação dos resultados anteriormente obtidos. Serão avaliados os gastos para armazenamento, inserção, remoção e busca de elementos.

Como um complemento ao acima avaliado, será analisado também uma otimização da árvore de busca binária, a árvore AVL e o algoritmo de rotação para balanceamento da mesma.

### 3 A ESTRUTURA

#### 3.1 DEFINIÇÃO

uma árvore binária de busca (ou árvore binária de pesquisa) é uma estrutura de dados de árvore binária baseada em nós, onde todos os nós da subárvore esquerda possuem um valor numérico inferior ao nó raiz e todos os nós da subárvore direita possuem um valor superior ao nó raiz. (WIKIPEDIA, )

#### 3.2 ELEMENTOS E TERMINOLOGIAS

- Nós - são todos os itens guardados na árvore
- Raiz - é o nó do topo da árvore
- Filhos - são os nós que vem depois dos outros nós
- Pais - são os nós que vem antes dos outros nós
- Folhas - são os nós que não têm filhos; são os últimos nós da árvore
- Grau - número de sub-árvores relacionadas àquele nó.
- Altura - comprimento do caminho mais longo do nó a uma folha
- Profundidade - comprimento do unico caminho entre o nó raiz e o nó avaliado

## 4 DISCUSSÕES E RESULTADOS

A árvore de busca binária é uma estrutura de dados focada em armazenar elementos de forma ordenada afim de agilizar o máximo possível a tarefa de busca. Contudo, para garantir essa melhora na eficiencia são necessários alguns pré-requisitos, além de dificultar de certa forma outras operações.

O ideal para uma árvore binária é garantir um tempo de  $O(n \log(n))$  ou algo próximo disso, e para isso é preciso uma árvore o mais balanceada possível, isto é,

Ao inserir elementos na árvore deve-se tomar cuidado com o elemento inicial e na ordenação relativa desses, pois pode-se facilmente degenerar a estrutura da árvore para uma lista, ou algo próximo disso, acabando completamente com seu propósito.

O armazenamento na árvore binária tem um custo semelhante a uma lista dinâmica encadeada, ocupando um espaço maior do que uma lista estática comum, porém com a vantagem de não ocupar espaço desnecessário e poder aumentar a lista a qualquer momento.

A inserção tem um custo baixo, devido a organização com ponteiros, é necessário apenas uma atribuição após encontrar o lugar, garantindo algo próximo de  $O(\log(n))$ , como será explicado a mais a frente.

A remoção nessa estrutura, apesar de não ter um custo alto, tem uma grande complexidade e pode acabar com o balanceamento da árvore, prejudicando assim todas as outras funcionalidades.

A busca na árvore pode ser realizada como se estivesse em uma lista estática ordenada realizando uma busca binária, porém ocorre de forma ainda mais direta, devido sua organização. Por conta disso é quase sempre possível conseguir um tempo  $O(\log(n))$ , porém, assim como todas operações, isso depende do balanceamento.

Uma árvore AVL é uma árvore binária de busca balanceada, e isso implica que a diferença na altura das duas sub-árvores filhas diferir em módulo de até uma unidade. Garantindo uma árvore ser AVL, garante-se uma altura de  $\log(n)$  e também um custo de  $O(\log(n))$  para busca inserção e remoção.

Um método utilizado para garantir o balanceamento é um algoritmo de rotação executado conforme a necessidade após operações de remoção ou inserção.

De forma superficial, existem quatro tipos de rotações, à direita ou à esquerda, e simples ou dupla, que quando executados em conjunto, literalmente rotacionam uma sub-árvore para balanceá-la.



## REFERÊNCIAS

WIKIPEDIA. **Árvore binária de busca** - **Wikipedia**. Disponível em: <[https://pt.wikipedia.org/wiki/Árvore\\_binária\\_de\\_busca](https://pt.wikipedia.org/wiki/Árvore_binária_de_busca)>. Acesso em: 19 set. 2018.