

**UNIVERSIDADE EDUARDO MONDLANE**

**FACULDADE DE ENGENHARIA**

**DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA**

**Discentes**: Shimwa, Jean

Dava, Rafael

Cumbane, Malingas

Dima, Valdemiro

Zengeni, Kevin

Árvores Binárias

**ESTRUTURA DE DADOS E ALGORITMOS**

V Grupo

Maputo, Setembro de 2022

**Introdução**

Uma árvore é um tipo específico de grafo onde cada elemento designado por **nó**, tem zero ou mais sucessores, mas apenas um predecessor (excepto o primeiro **nó**, a que se dá o nome de **raíz** da árvore). Neste trabalho vamos dar mais atenção as árvores binárias, por serem o tipo de árvore mais utilizado na computação, onde a principal utilização destas árvores binárias são as árvores de busca.

**Objectivos**

**Gerais:**

* Estudo de árvores binárias.

**Específicos:**

* Ter a capacidade de aplicar corretamente a estrutura de dados de árvores binárias;
* Realizar diversas operações nas árvores binárias;

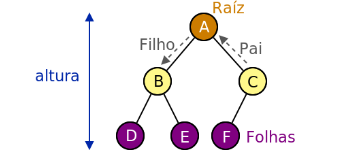
**Árvores Binárias**

**Árvores**

Árvore binária é uma estrutura de dados caracterizada por ter um elemento denominado raiz, com duas estruturas diferentes, denominadas sub-árvore esquerda e sub-árvore direita; ou por não ter elemento algum (Árvore vazia).

Percebe-se que a definição é recursiva e, devido a isso, muitas operações sobre árvores binárias utilizam a recursividade.

**Terminologia**



* Ao predecessor (único) de um nó, chamamos **pai;**
* Os sucessores de um nó são os seus **filhos;**
* O **grau** de um nó é o seu número de filhos
* Uma **folha** é um nó sem filhos, ou seja , de grau 0;
* A **raiz** é o único nó sem pai
* Uma **sub-árvore** é um subconjunto de nós (ligados) da árvore;
* Os arcos que ligam os nós, chamam-se **ramos**;
* Chama-se **caminho** á sequência de ramos entre dois nós;
* A **profundidade** de um nó é o comprimento do caminho desde a raiz até esse nó (a profundidade da raiz é zero)
* A **altura** de uma árvore é a profundidade máxima de um nó da árvore;
* A **aridade** de uma árvore é o grau máximo de um nó;
* Uma **árvore binária** é uma árvore de aridade 2, isto é, cada nó possui no máximo dois filhos, designados por filho **esquerdo** e **direito.**

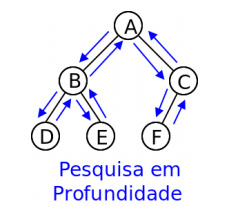
**Escrita de Nós de uma Árvore**

Para escrever o conteúdo de um árvore temos que passar por todos nós utilizando uma ordem:

**Pesquisa em Profundidade (DFS:** depth-first-search**)**

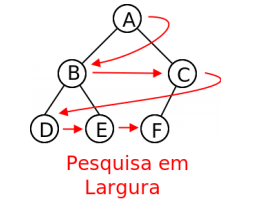
Nesta pesquisa visita-se todos os nós da sub-árvore de um filho antes de visitar a sub-árvore do outro filho. Existem três ordens nesta pesquisa que são:

* **PreOrder:**  raíz escrita antes das duas sub-árvores**;**
* **InOrder:** raíz escrita entre as duas sub-árvores**;**
* **PostOrder:** raíz escrita depois das duas sub-árvores**.**

****

**Pesquisa em Largura (BFS:** breadth-first-search**)**

Nesta pesquisa visita-se os nós por ordem crescente de profundidade

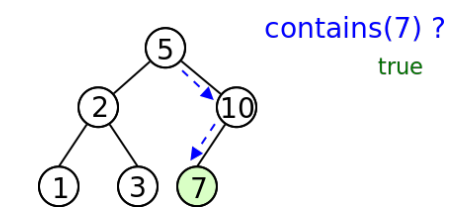


**Leitura de Árvores**

Para leitura de árvores, pode ser utilizada umas das ordens acima mencionadas.

**Pesquisa de Valores na Árvore**

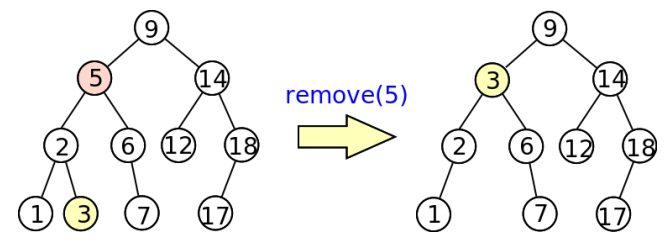
Percorre-se a árvore, partindo do ramo esquerdo ou direito consoante o valor (se é menor ou maior que o do nó “actual”)



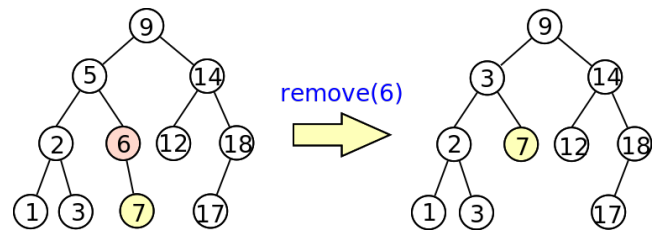
**Remoção de um valor**

Para remover um valor da árvore, percorre-se toda árvore até achar o nó com o valor e retirar. A seguir :

* Se o nó retirado tinha um ramo filho, basta “subir” esse filho até à posição correspondente;
* Se tiver dois ramos filhos, os candidatos a ficarem nessa posição são:
* O maior nó do ramo esquerdo



* O menor nó do ramo direito.



**Vantagens e Desvantagens**

**Vantagens:**

* Operações de pesquisa em árvores binárias são muito rápidas;
* A representação da árvore binária é muito simples de representar e entender;
* É fácil chegar ao nó filho através do pai e vice-versa;
* Simples de implementar;
* Têm uma estrutura hierárquica;
* Fácil inserção de dados comparado a outra base de dados;
* Possível adicionar uma infinidade de nós.

**Desvantagens**

* Remoção de nó não fácil;
* A operação de acesso em árvore binária é mais lenta do que em arrays;
* Muita área de memória desperdiçada;
* Alto tempo de execução quando feita uma alteração na lista;

**Conclusão**

Depois do estudo,representação e implementação de uma árvore binária, é possível notar/perceber/concluir que além de sua estrutura hierárquica que facilita na organização dos dados e a leitura dos mesmos, a estrutura é muito útil para casos de decisão bidirecionais.

**Bibliografia**

- [www.ime.usp.br](http://www.ime.usp.br): Árvores binárias ;

- [www.dcc.fc.up.pt](http://www.dcc.fc.up.pt) : Árvores Binárias;

- <https://pt.m.wikibooks.org>: Algoritmos e Estruturas de Dados/ Árvores Binárias;

- [www.devmedia.com.br](http://www.devmedia.com.br) : Trabalhando com árvores binárias em Java;

- [www.geeksforgeeks.org/applications-advantages-and-disadvantages-of-binary-tree , www.quora.com/what-is-the-advantage-and-disadvantage-of-a-binary-tree:](http://www.geeksforgeeks.org/applications-advantages-and-disadvantages-of-binary-tree:) Vantagens e desvantagens de implementação de árvores binárias