

Árvores Binárias

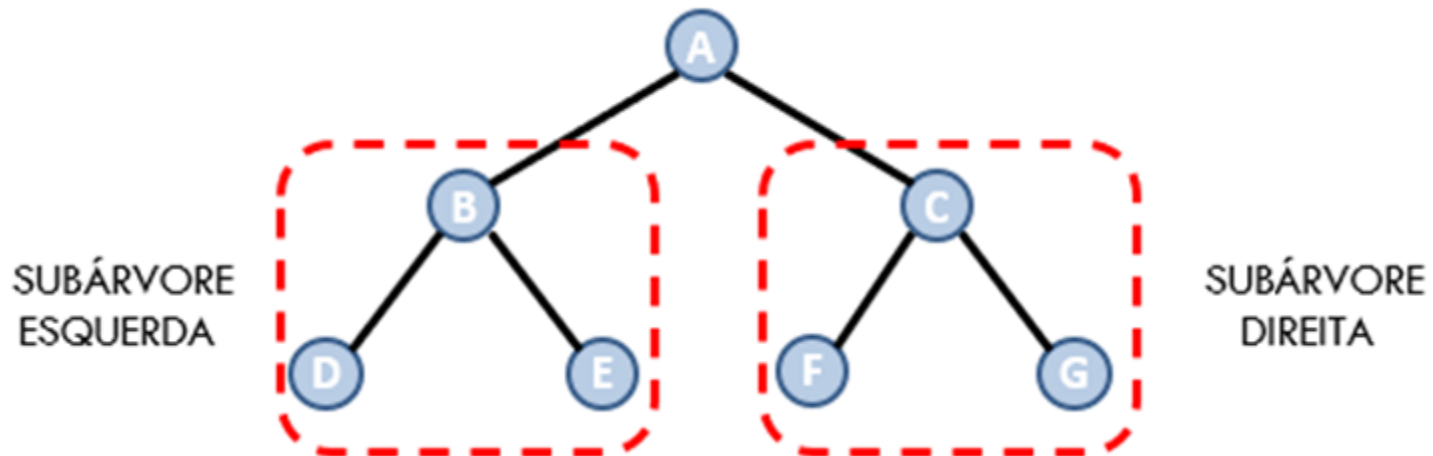
Disciplina: Estruturas de Dados 2– ED43S
Curso: Tecnologia em Análise e Desenvolvimento de Sistemas

Departamento Acadêmico de Informática (Dainf)

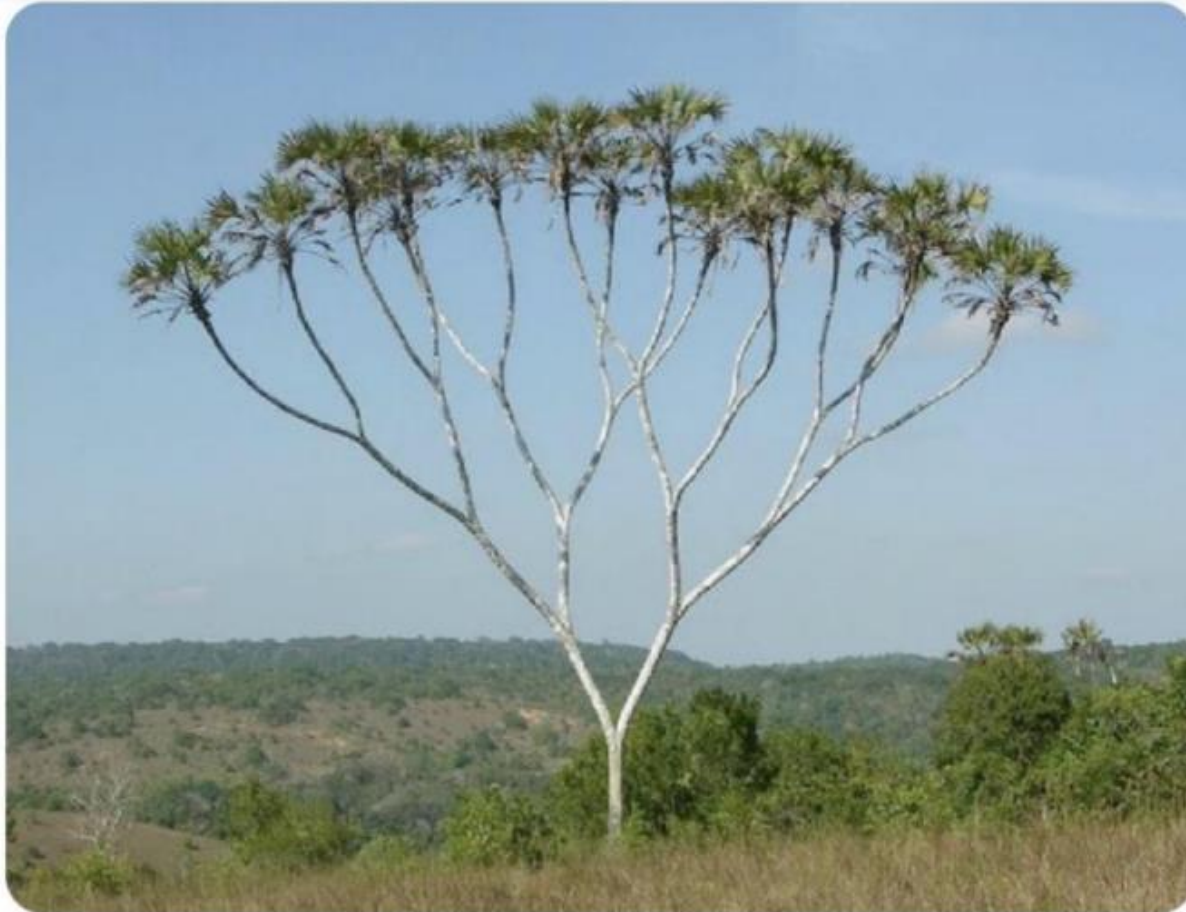
Professora: Rúbia Eliza de Oliveira Schultz Ascari
rubia@utfpr.edu.br

Árvore Binária

- É um tipo especial de árvore
 - Cada nó pode possuir nenhuma, uma ou no máximo duas **subárvores**
 - Subárvore da **esquerda** e a da **direita**
 - Usadas em situações onde, a cada passo, é preciso tomar uma decisão entre duas direções



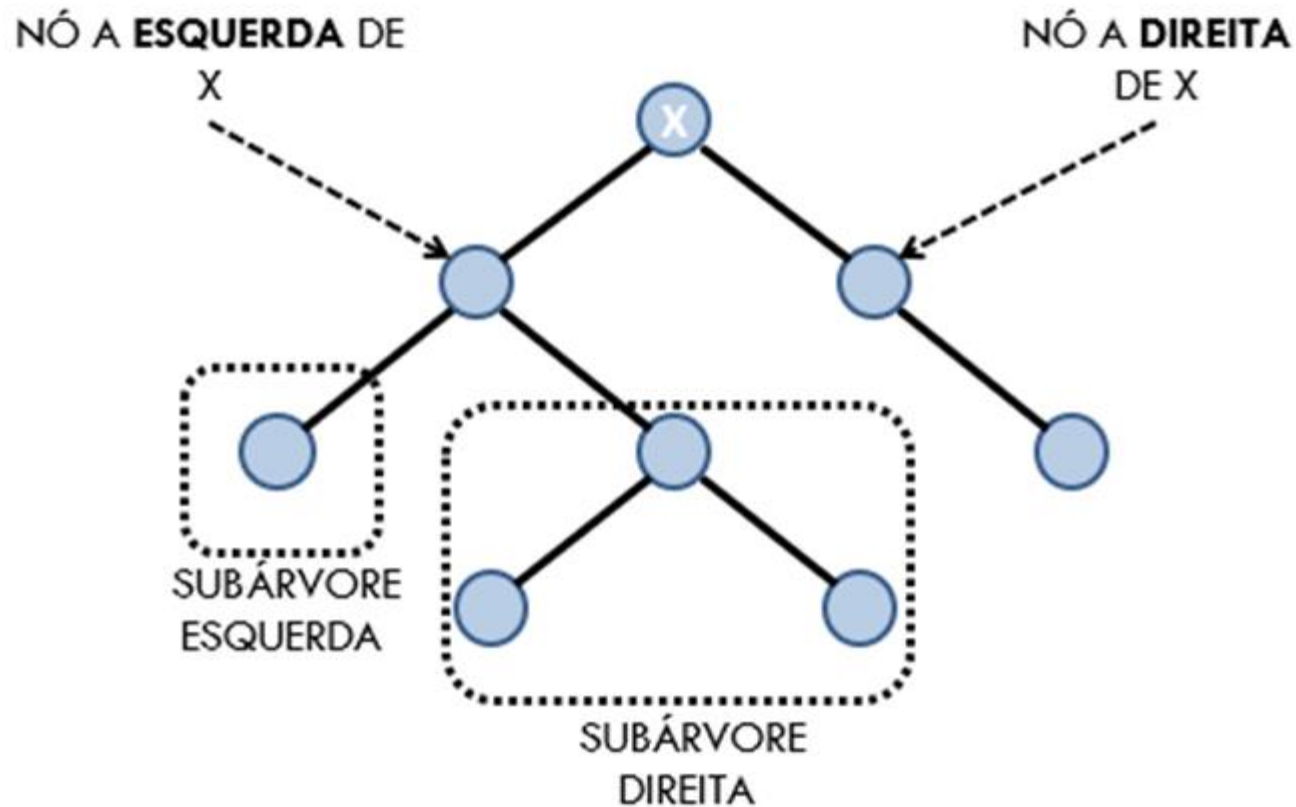
Árvore Binária



Árvore binária na natureza

Árvore Binária

- Exemplo de árvore binária:



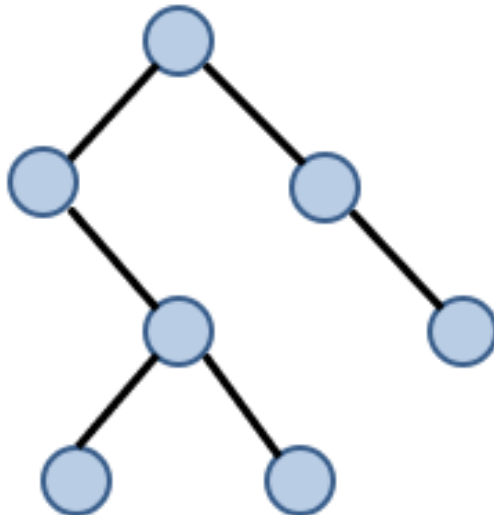
Árvore Binária

- Existem alguns tipos de árvores binárias
 - Estritamente binária
 - Completa
 - Cheia
 - Zigue Zague

Árvore Binária

- Árvore estritamente binária
 - Cada nó possui sempre ou 0 (no caso de nó folha) ou 2 subárvores
 - Nenhum nó tem **filho único**

ÁRVORE
BINÁRIA

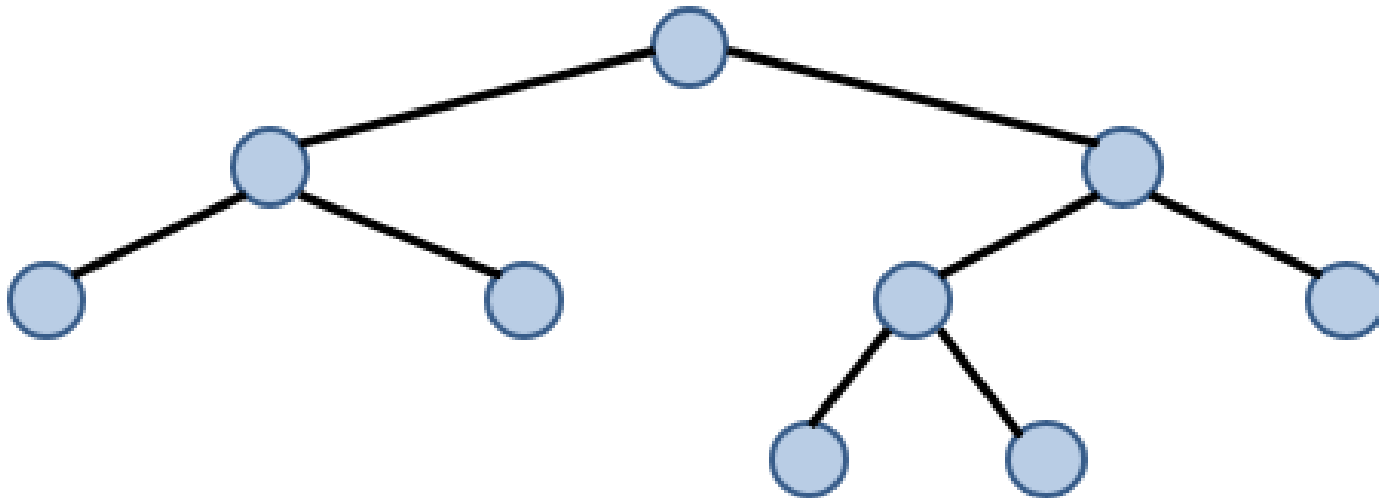


ÁRVORE ESTRITAMENTE
BINÁRIA



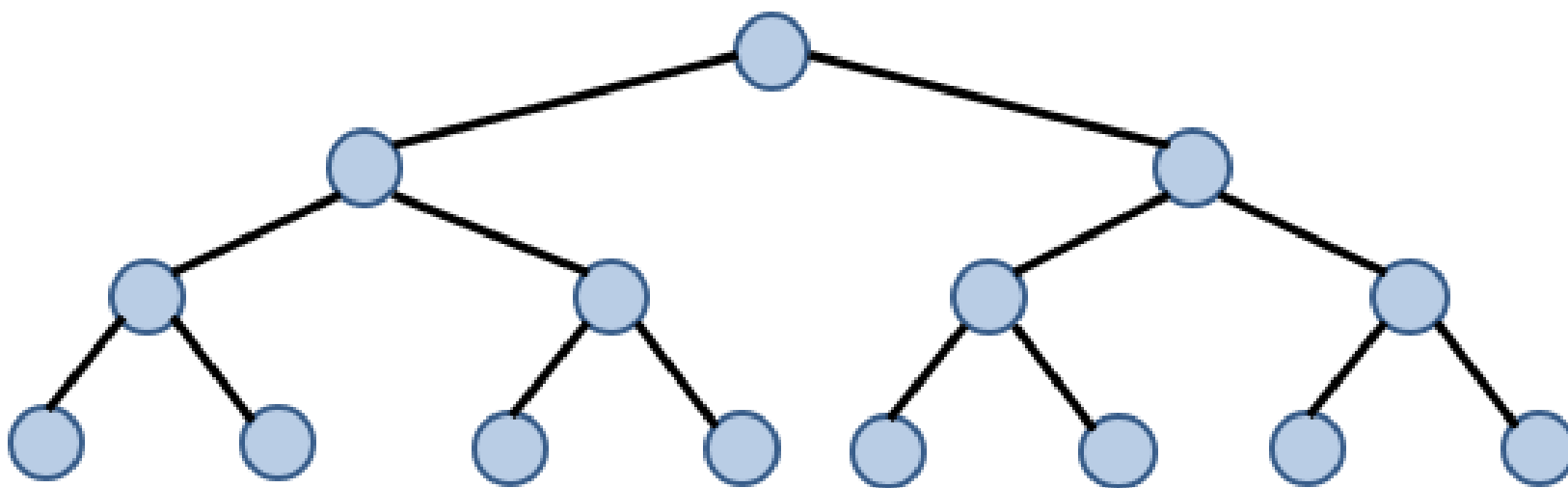
Árvore Binária

- Árvore binária completa
 - A diferença de altura entre as subárvores de qualquer nó é no máximo 1
 - Se a altura da árvore é **D**, cada nó folha está no nível **D** ou **D-1**



Árvore Binária

- Árvore binária cheia
 - Árvore estritamente binária onde todos os nós folhas estão no mesmo nível



Nível (n)	0	1	2	3
Nº de nós	1	2	4	8
Potência	2^0	2^1	2^2	2^3

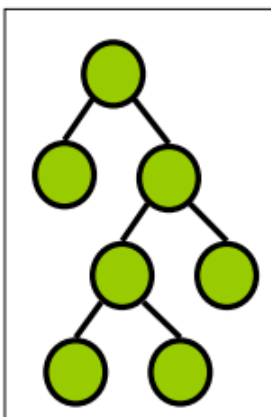
Árvore Binária

- Árvore zigue zague
 - Cada nó possui apenas **filho único**, ou seja, os nós interiores possuem exatamente uma subárvore vazia.



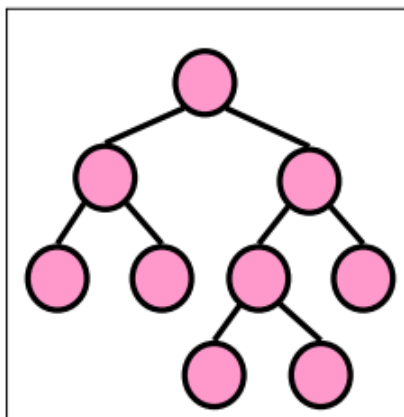
Árvore Binária

TIPOS DE ÁRVORES BINÁRIAS



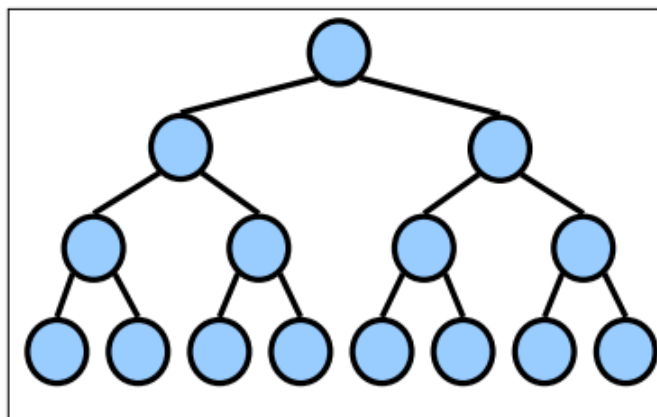
Estritamente Binária

0 ou 2 filhos



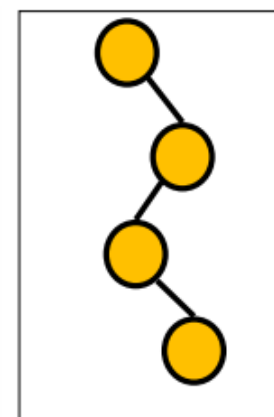
Binária Completa

Sub-árvores vazias apenas no último ou penúltimo nível



Binária Cheia

Sub-árvores vazias somente no último nível

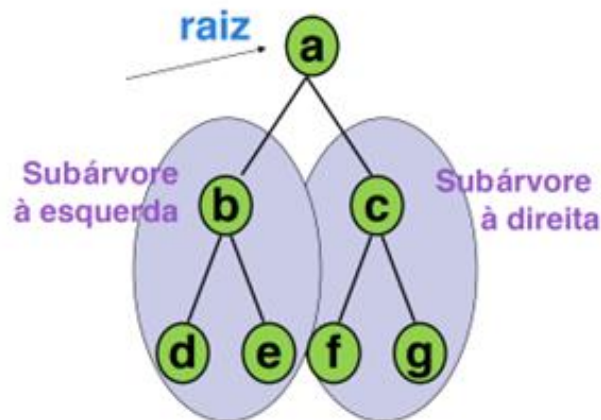


Zigue Zague

Nós internos com 1 subárvore vazia

Árvore Binária

- Percorrer todos os nós é uma operação muito comum em árvores binárias
 - Cada nó é visitado uma única vez
 - Isso gera uma sequência linear de nós, cuja ordem depende de como a árvore foi percorrida
 - Não existe uma **ordem natural** para se percorrer todos os nós de uma árvore binária
 - A visita ao nó pode ser feita para executar alguma ação em cada nó
 - Essa ação pode ser mostrar (imprimir) o valor do nó, modificar esse valor, etc.



Árvore Binária

- Percursos em árvores binárias:
 - Um percurso (ou caminhamento) define uma sequência de nós
 - Cada nó passa a ter um nó seguinte, ou um nó anterior, ou ambos (exceto árvore com 1 só nó)
 - A Sequência de nós depende do percurso adotado.

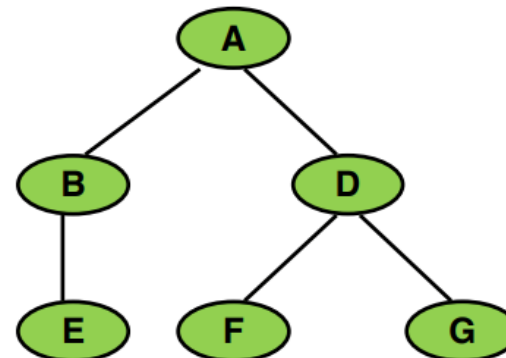
Exemplo

- Percurso 1:

A – B – D – E – F – G

- Percurso 2:

A – B – E – D – F – G



Árvore Binária

- Percursos comuns em árvores binárias:

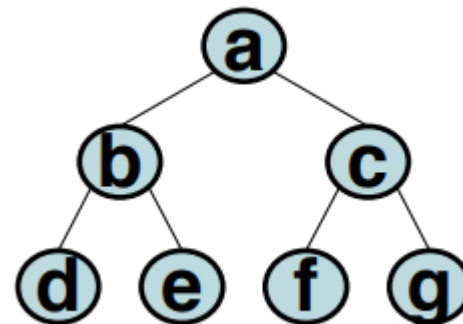
- Pré-Ordem (Profundidade)

- 1) Visita a **raiz**
- 2) Percorre a sub-árvore **esquerda**
- 3) Percorre a sub-árvore **direita**

Exemplo

- Percurso 1:

a - b - d - e - c - f - g



Árvore Binária

- Percursos comuns em árvores binárias:

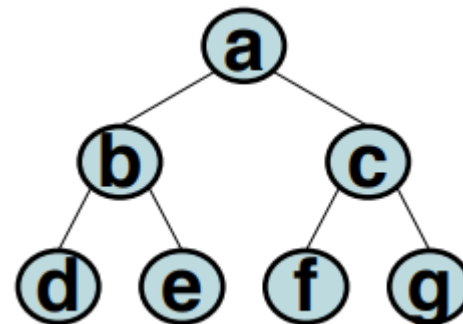
- Em-Ordem (Ordem Simétrica)

- 1) Percorre a sub-árvore **esquerda**
- 2) Visita a **raiz**
- 3) Percorre a sub-árvore **direita**

Exemplo

- Percurso 1:

d - b - e - a - f - c - g



Árvore Binária

- Percursos comuns em árvores binárias:

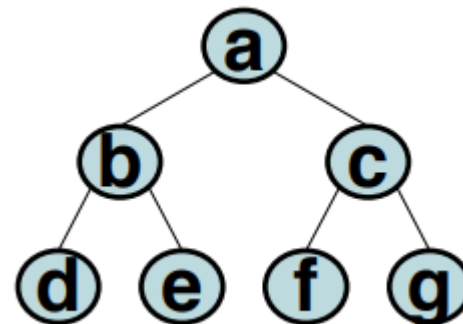
➤ Pós-Ordem

- 1) Percorre a sub-árvore **esquerda**
- 2) Percorre a sub-árvore **direita**
- 3) Visita a **raiz**

Exemplo

- Percurso 1:

d - e - b - f - g - c - a



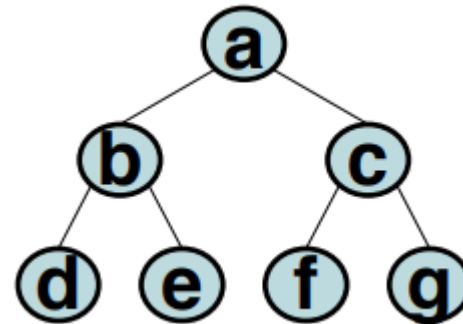
Árvore Binária

- Percursos comuns em árvores binárias:
 - Largura
 - 1) Visita é feita por nível, da **esquerda** para a **direita**

Exemplo

- Percurso 1:

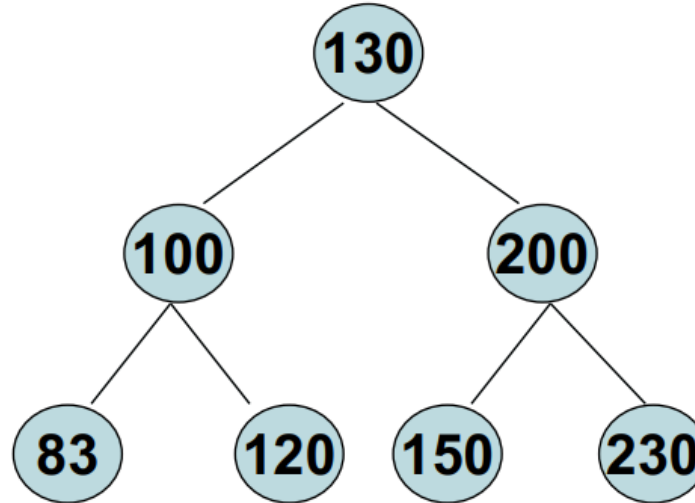
a - b - c - d - e - f - g



Árvore Binária

- Percursos comuns em árvores binárias:

➤ Exercício 1:

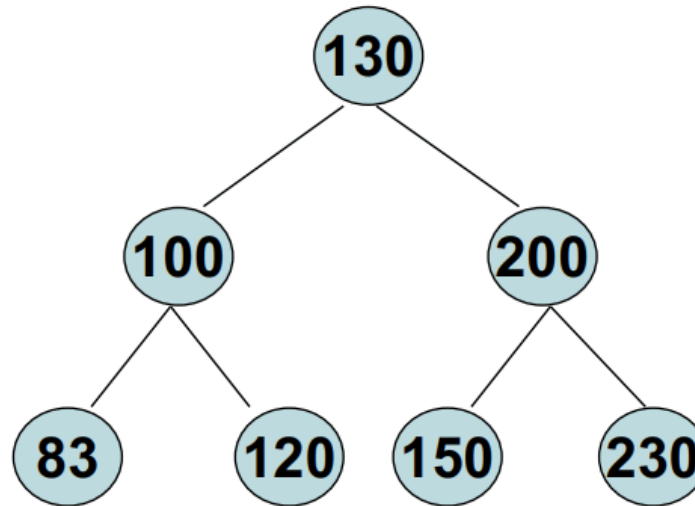


- Pré-ordem:
- Em-ordem:
- Pós-ordem:
- Largura:

Árvore Binária

- Percursos comuns em árvores binárias:

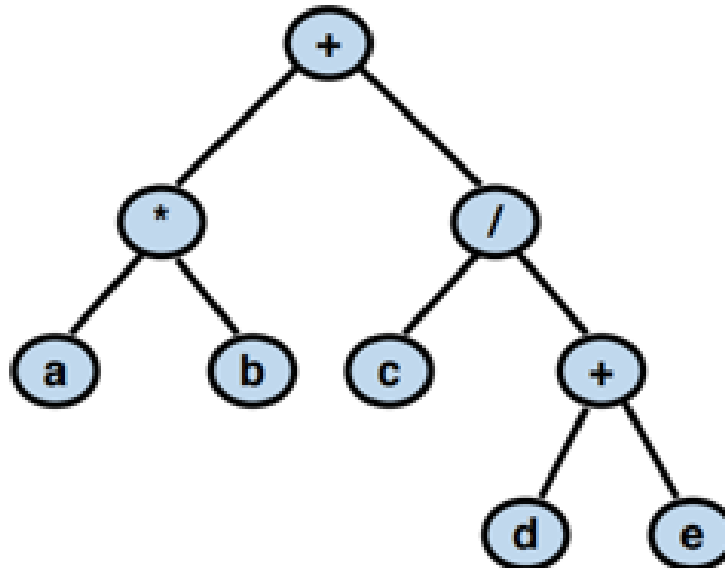
➤ Exercício 1:



- Pré-ordem: 130 – 100 – 83 – 120 – 200 – 150 – 230
- Em-ordem: 83 – 100 – 120 – 130 – 150 – 200 – 230
- Pós-ordem: 83 – 120 – 100 – 150 – 230 – 200 – 130
- Largura: 130 – 100 – 200 – 83 – 120 – 150 – 230

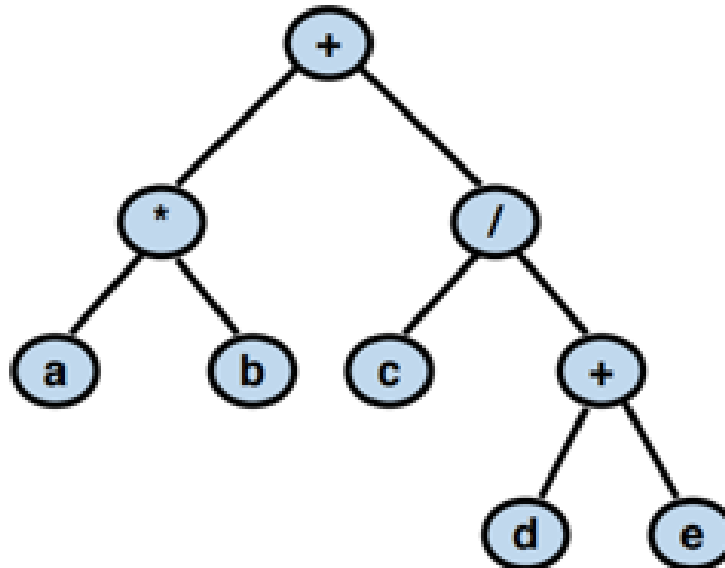
Árvore Binária

- Percursos comuns em árvores binárias:
 - Exercício 2: Qual o percurso mais indicado para montar a equação matemática armazenada na árvore binária a seguir, para que a leitura seja feita da forma como nós estamos acostumados a utilizar?



Árvore Binária

- Percursos comuns em árvores binárias:
 - Exercício 2: Qual o percurso mais indicado para montar a equação matemática armazenada na árvore binária a seguir, para que a leitura seja feita da forma como nós estamos acostumados a utilizar?



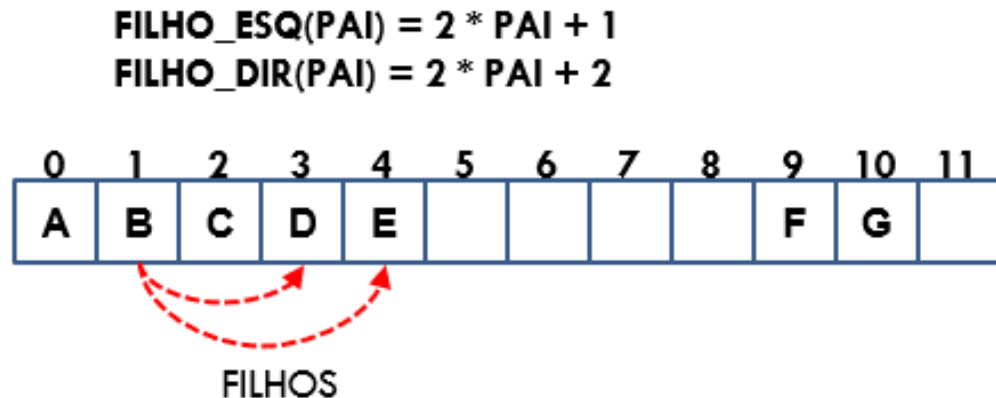
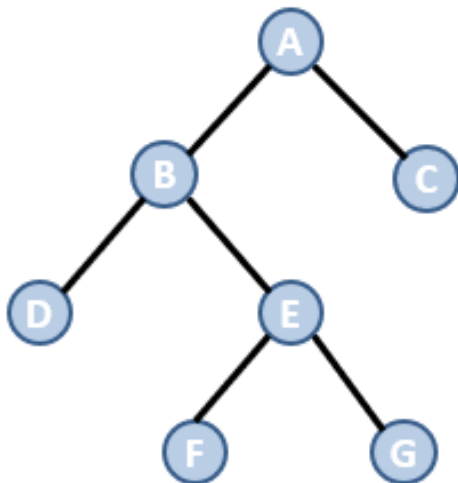
- Em ordem: $(a * b) + (c / (d + e))$

Árvore Binária

- Existem duas abordagens muito utilizadas para implementar uma árvore no computador:
 - Usando um array (alocação estática)
 - Usando uma lista encadeada (alocação dinâmica)

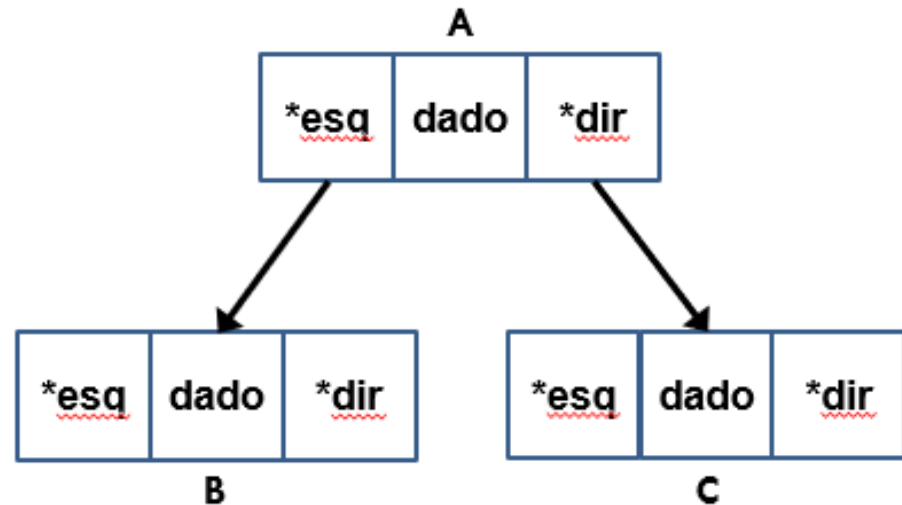
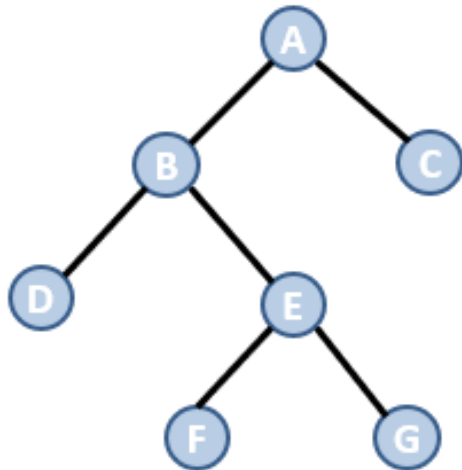
Árvore Binária

- Usando um array (alocação estática)
 - Necessário definir o número máximo de nós
 - Tamanho do array
 - Usa 2 funções para retornar a posição dos filhos à esquerda e à direita de um pai



Árvore Binária

- Lista encadeada (alocação dinâmica)
 - Espaço de memória alocado em tempo de execução
 - A árvore cresce à medida que novos elementos são armazenados, e diminui à medida que elementos são removidos



Árvore Binária

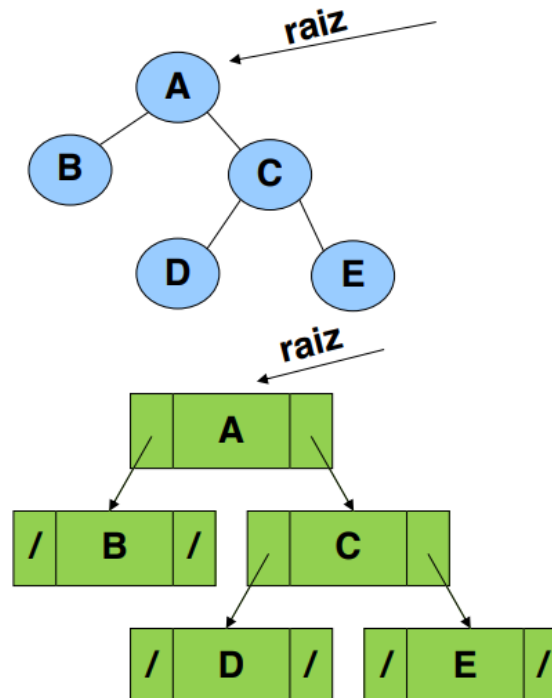
Um exemplo de implementação usando Lista Encadeada (TAD) está disponível para download no Moodle.

Fonte do algoritmo: <https://programacaodescomplicada.wordpress.com/complementar/>

```
struct NO{  
    int info;  
    struct NO *esq;  
    struct NO *dir;  
};
```

esq	info	dir

```
A  
--B  
----vazio  
----vazio  
--C  
----D  
-----vazio  
-----vazio  
----E  
-----vazio  
-----vazio
```



Árvore Binária de Busca

- Definição
 - É uma árvore binária
 - Cada nó pode ter 0, 1 ou 2 filhos
 - Cada nó da árvore possui um valor (chave) associado a ele
 - Não existem valores repetidos
 - Esse valor determina a posição do nó na árvore
- Regra para posicionamento dos valores na árvore
 - Para cada nó pai
 - todos os valores da subárvore **esquerda são menores** do que o nó pai
 - todos os valores da subárvore **direita são maiores** do que o nó pai;
 - Inserção e remoção devem ser realizadas respeitando essa regra de posicionamento dos nós.

Referências

- Conteúdo baseado no material elaborado pelo professor André Backes. Disponível em <https://programacaodescomplicada.wordpress.com/complementar/>.
- Backes, André Ricardo, Estruturas de dados descomplicada: em linguagem C, 1ª ed., Rio de Janeiro: Elsevier, 2016.
- Braganholo, Vanessa. Estruturas de Dados e Seus Algoritmos – Árvores e Árvores Binárias. Disponível em <http://www2.ic.uff.br/~vanessa/material/ed/03-ArvoresBinarias.pdf>. Acesso em 05/05/2021.

Dúvidas

- ???