

# ÁRVORES BINÁRIAS

Prof. Muriel Mazzetto  
Estrutura de Dados

# Árvores n-árias

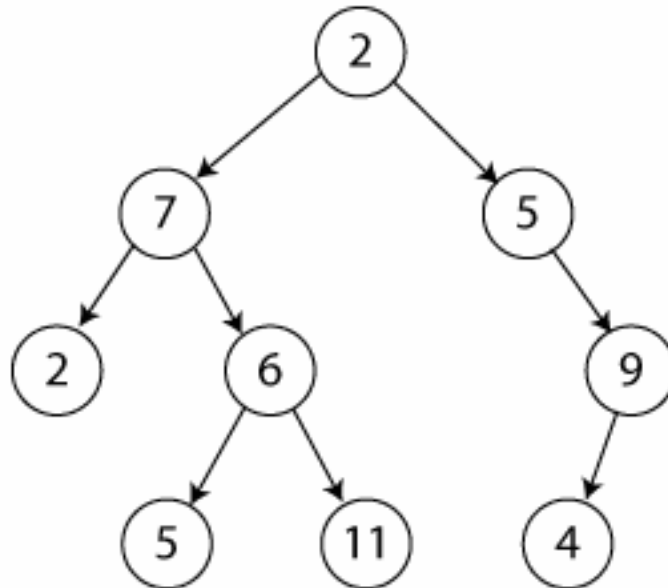
2

- São árvores que restringem em  $n$  a quantidade de filhos que cada nó pode ter (grau do nó).
- Normalmente implementadas pela estrutura de *nós encadeados*.
- A mais popular e mais utilizada é a árvore com o valor de  $n = 2$ , denominada de **árvore binária**.

# Árvore binária

3

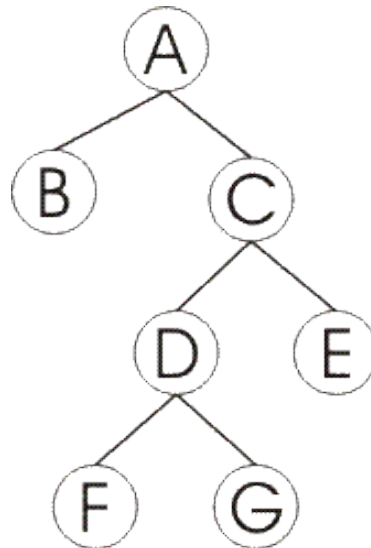
- Uma estrutura de dados do tipo árvore **ordenada**.
- Cada nó indica o filho à **esquerda** e o filho à **direita**.



# Árvore binária

4

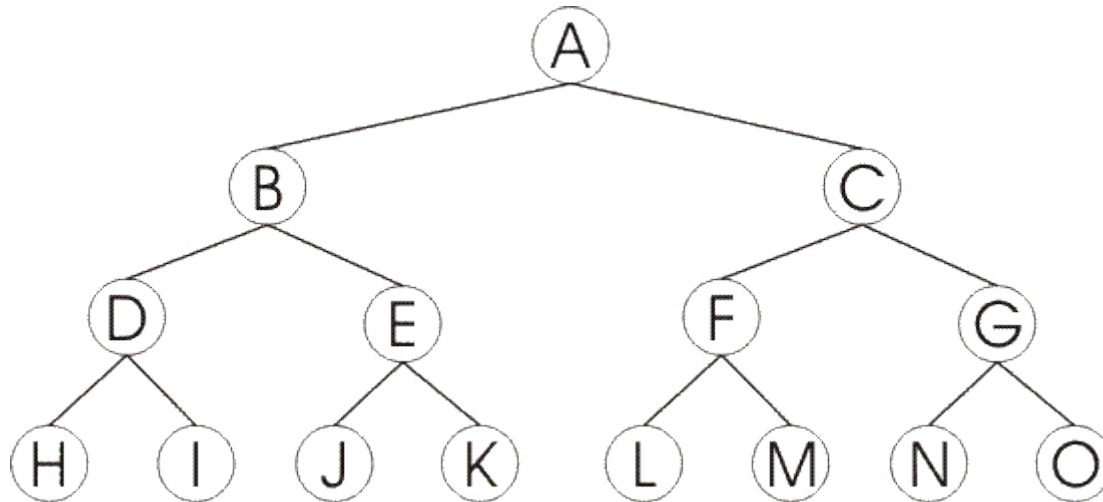
- **Árvore Estritamente Binária:** todos os nós possuem zero (nenhum) ou dois filhos.
- Todo nó interno (não folha) possui dois filhos.



# Árvore binária

5

- **Árvore Binária Completa:** é estritamente binária e possui um nível  $x$  definido.
- Todos os seus nós folha estão no mesmo nível  $x$ .

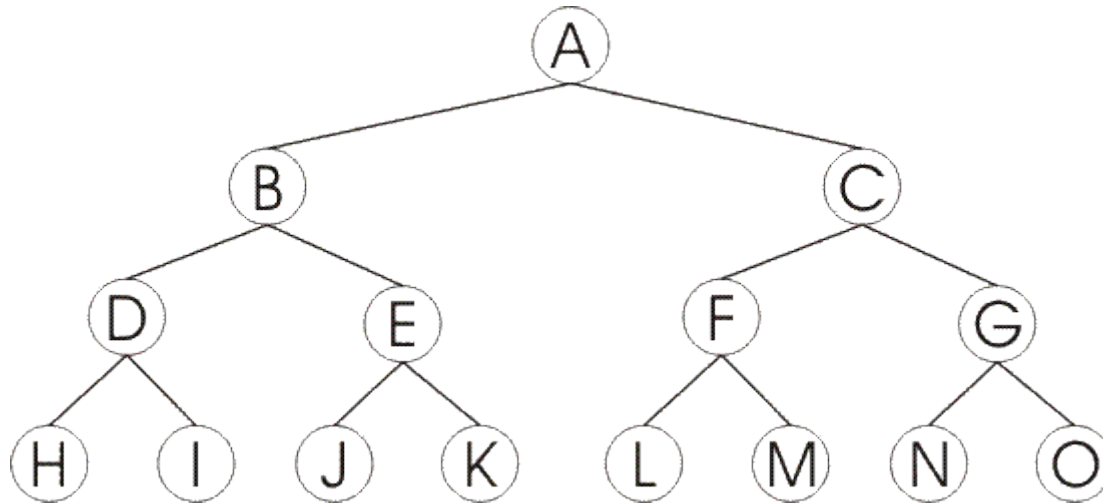


# Árvore binária

6

- A quantidade  $N$  de nós de uma **Árvore Binária Completa** pode ser determinada a partir da altura  $h$ , pela equação:

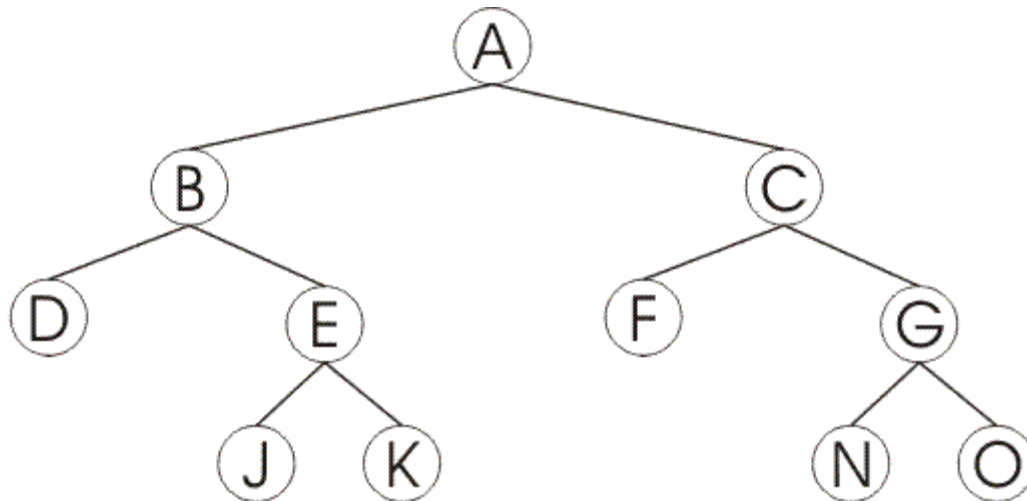
$$N = 2^h - 1$$



# Árvore binária

7

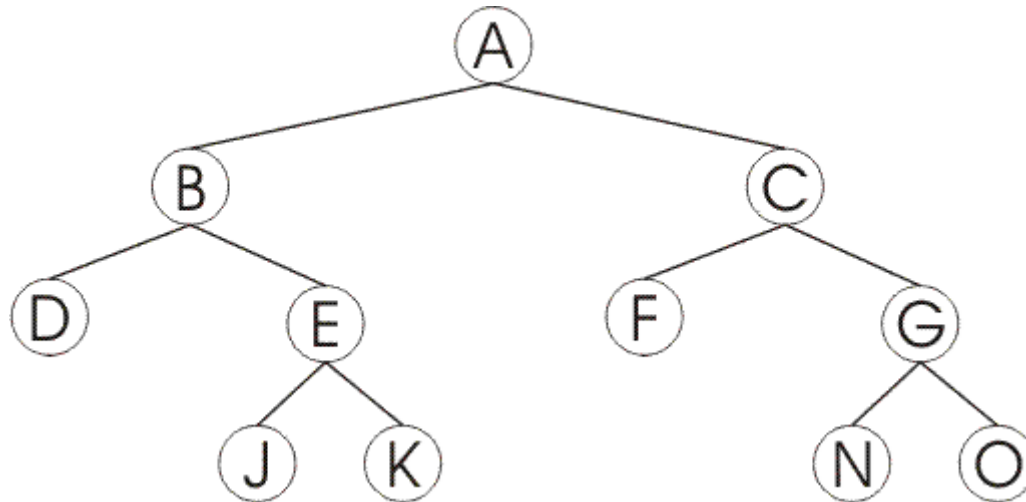
- **Árvore Binária Quase Completa:** se a diferença de altura entre as subárvores de qualquer nó é no máximo 1.
- As folhas devem estar no nível  $d$  ou  $d-1$ , para uma árvore de altura  $d$ .



# Árvore binária

8

- **Árvore Binária Balanceada:** manter a menor altura possível da árvore binária.
- Segue o padrão de uma Árvore Binária Quase Completa.

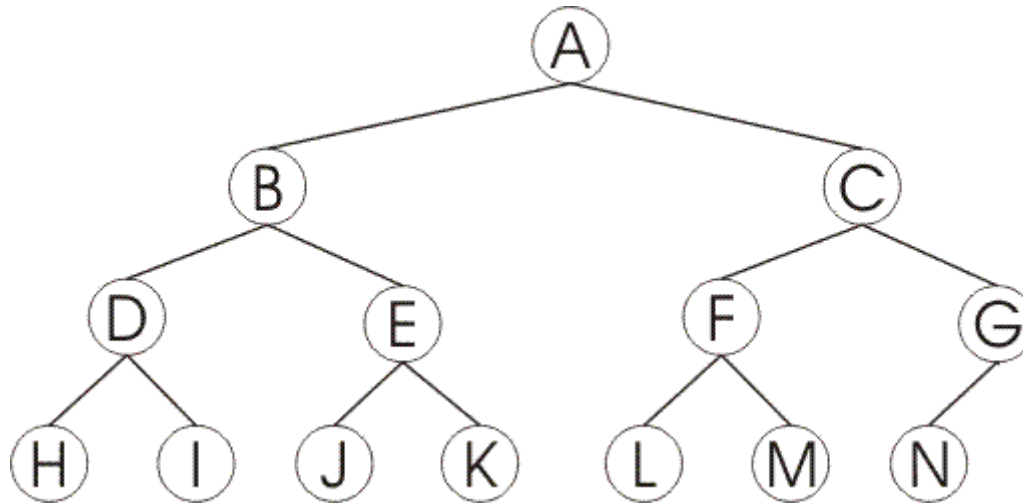




# Árvore binária

9

- **Árvore Binária Perfeitamente Balanceada:** para cada nó, o número de nós de suas subárvores esquerda e direita diferem em, no máximo, um.



# Percurso em Árvore Binária

10

- Forma sistemática de percorrer uma árvore.
- Visitar cada nó uma única vez no percurso, mantendo a ordem.
- Formas de percorrer árvores binárias:
  - ▣ Pré-ordem;
  - ▣ Em-ordem;
  - ▣ Pós-ordem.
- A “visita” ao nó consiste em executar alguma operação sobre os seus dados.

# Percurso em Árvore Binária

11

## □ **Pré-ordem:**

- Visitar a raiz (nó atual).
  - Percorrer a sua subárvore esquerda em pré-ordem.
  - Percorrer a sua subárvore direita em pré-ordem.
- Conhecido também como percurso em profundidade.

# Percurso em Árvore Binária

12

## □ **Em-ordem:**

- Percorrer a sua subárvore esquerda em em-ordem.
  - Visitar a raiz (nó atual).
  - Percorrer a sua subárvore direita em em-ordem.
- Conhecido também como percurso simétrico.

# Percurso em Árvore Binária

13

## □ Pós-ordem:

- Percorrer a sua subárvore esquerda em pós-ordem.
- Percorrer a sua subárvore direita em pós-ordem.
- Visitar a raiz (nó atual).

# Percurso em Árvore Binária

14

1- Pré-ordem: Raiz Esquerda Direita;

2- Em-ordem: Esquerda Raiz Direita;

3- Pós-ordem: Esquerda Direita Raiz;

□ Realizar a operação sempre que ocorrer a visita na Raiz.

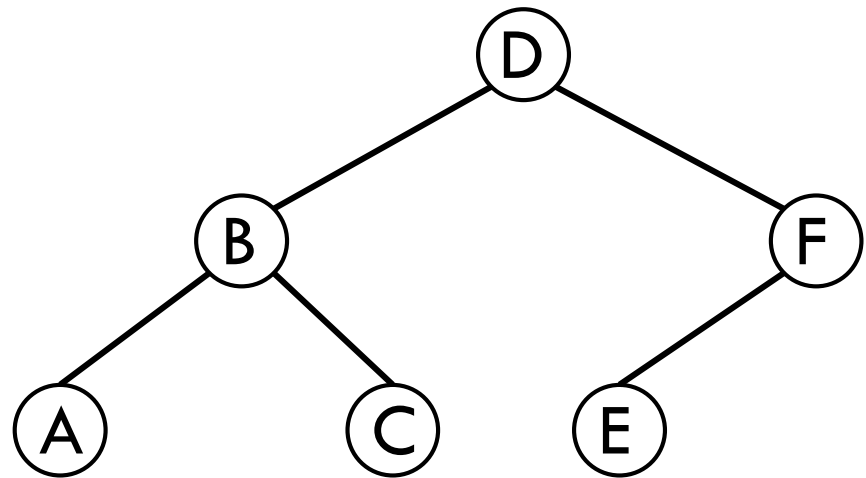
# Percurso em Árvore Binária

15

1- Pré-ordem: R E D;

2- Em-ordem: E R D;

3- Pós-ordem: E D R;



# Percurso em Árvore Binária

16

1- Pré-ordem: **R** **E** **D**;

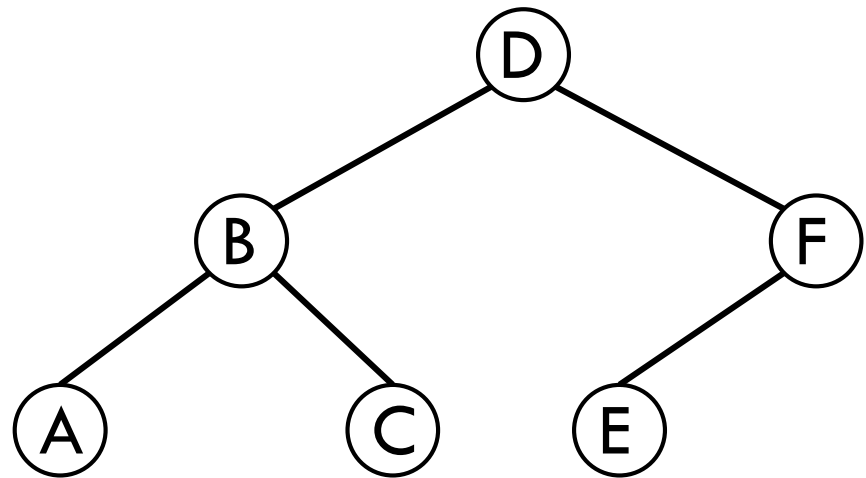
2- Em-ordem: **E** **R** **D**;

3- Pós-ordem: **E** **D** **R**;

1- D-B-A-C-F-E

2- A-B-C-D-E-F

3- A-C-B-E-F-D





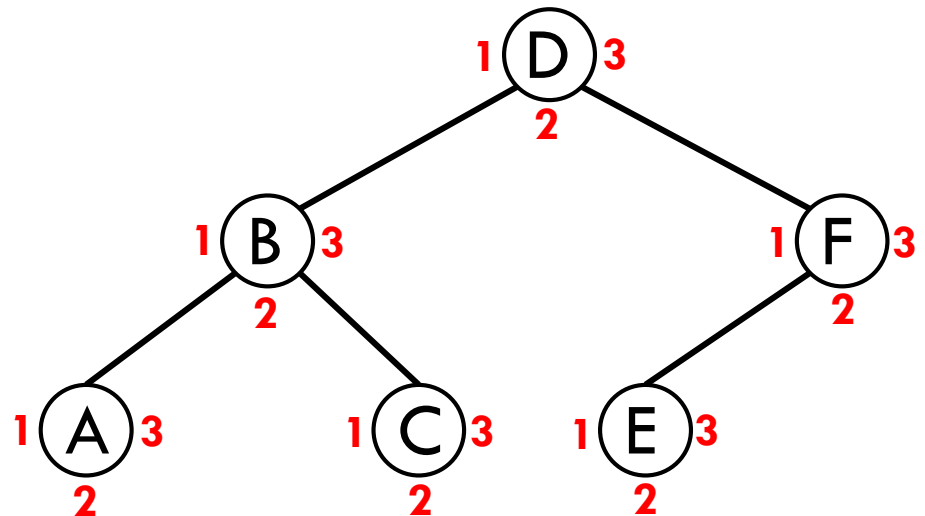
# Percurso em Árvore Binária

17

1- Pré-ordem: R E D;

2- Em-ordem: E R D;

3- Pós-ordem: E D R;



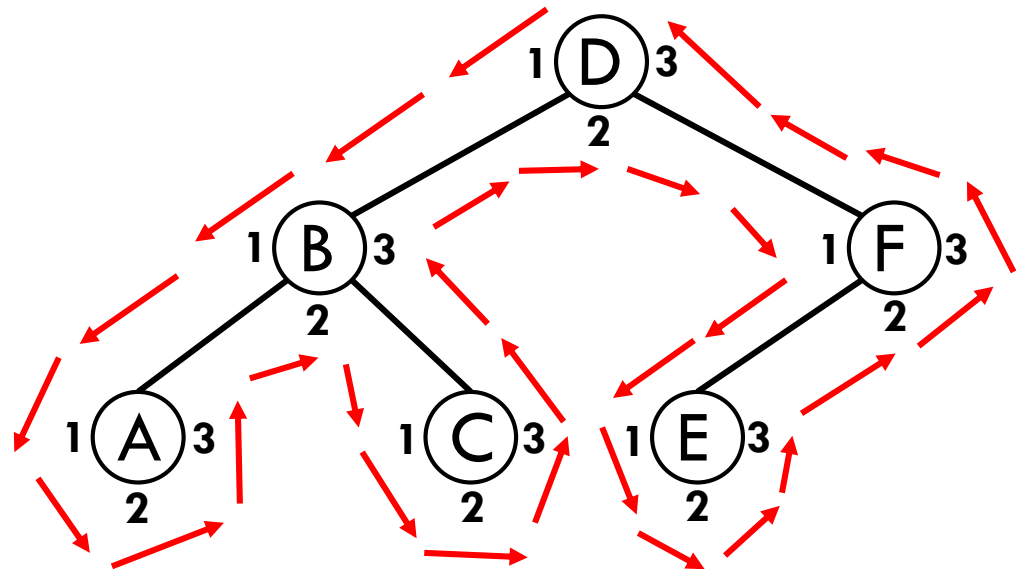
# Percurso em Árvore Binária

18

1- Pré-ordem: **R** **E** **D**;

2- Em-ordem: **E** **R** **D**;

3- Pós-ordem: **E** **D** **R**;



# Percurso em Árvore Binária

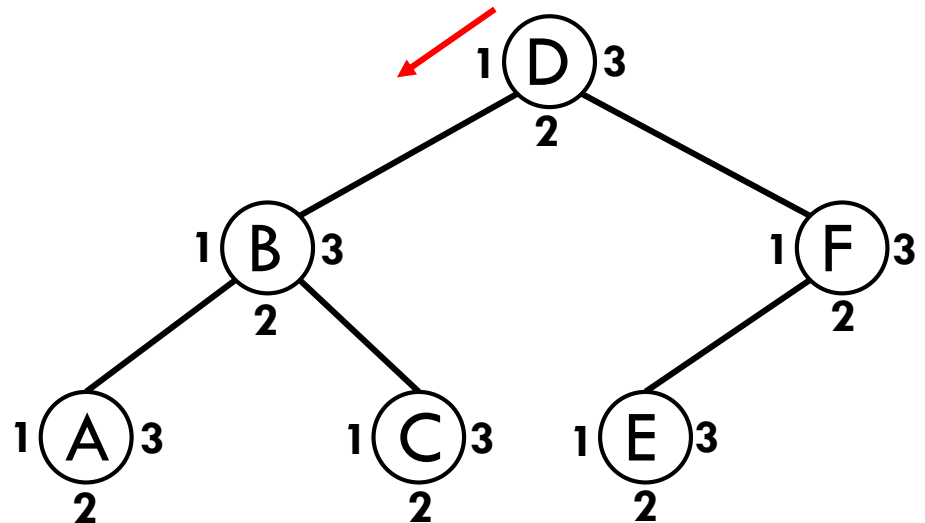
19

1- Pré-ordem: R E D;

2- Em-ordem: E R D;

3- Pós-ordem: E D R;

2-



# Percurso em Árvore Binária

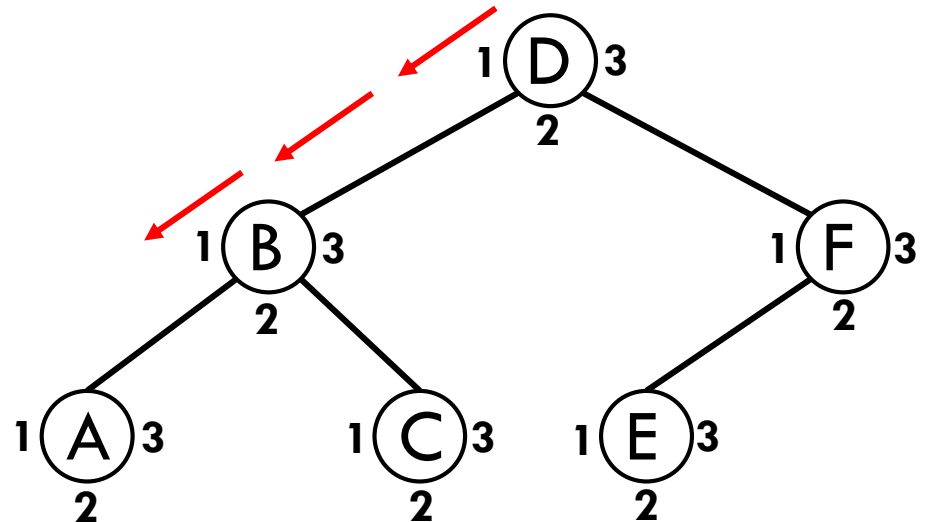
20

1- Pré-ordem: R E D;

2- Em-ordem: E R D;

3- Pós-ordem: E D R;

2-



# Percurso em Árvore Binária

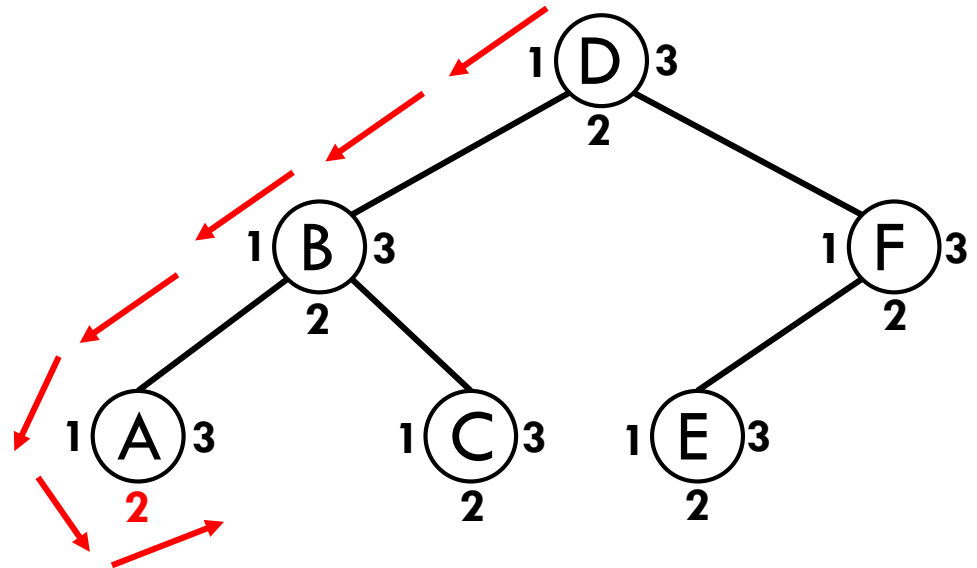
21

1- Pré-ordem: **R** **E** **D**;

2- Em-ordem: **E** **R** **D**;

3- Pós-ordem: **E** **D** **R**;

2- **A**



# Percurso em Árvore Binária

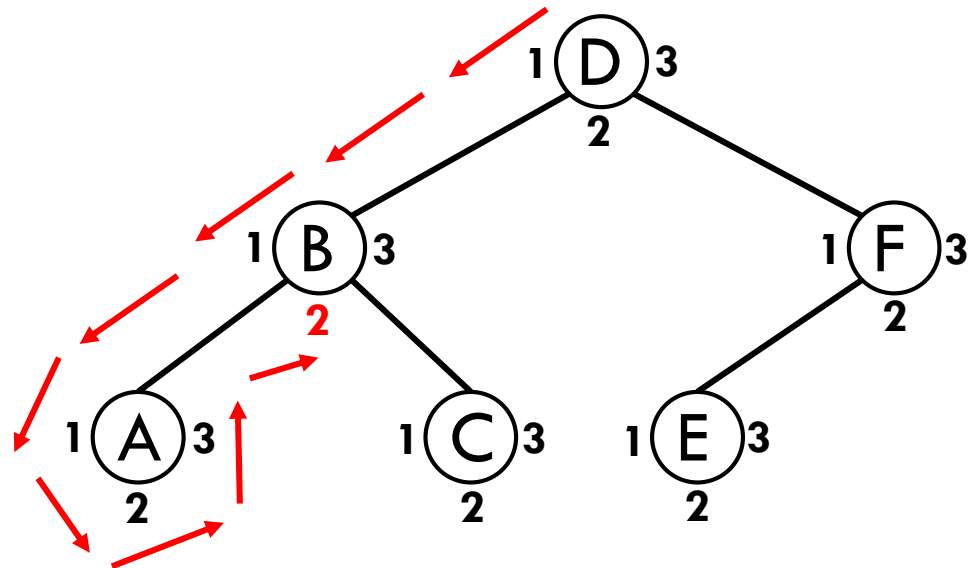
22

1- Pré-ordem: **R** E D;

2- Em-ordem: E **R** D;

3- Pós-ordem: E D **R**;

2- A-**B**



# Percurso em Árvore Binária

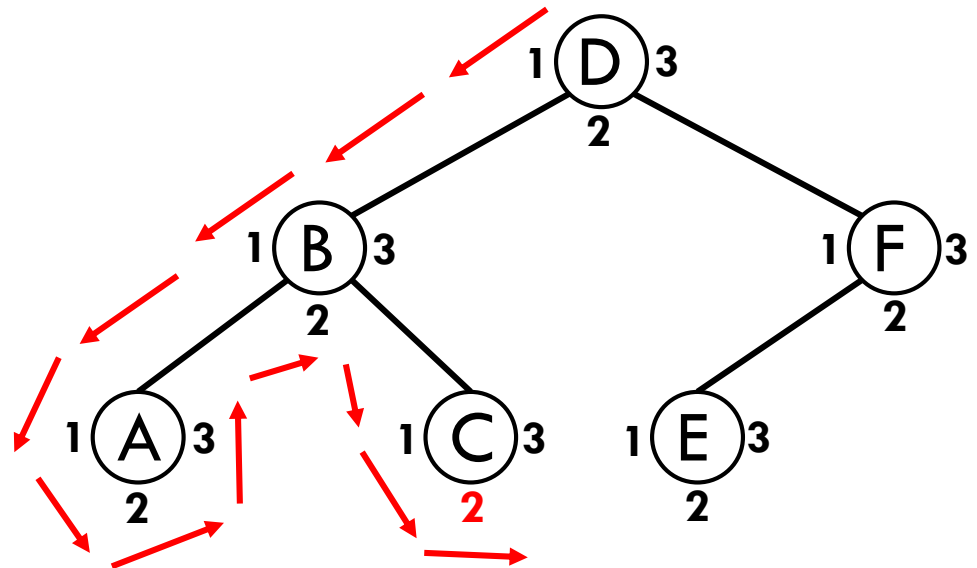
23

1- Pré-ordem: **R** **E** **D**;

2- Em-ordem: **E** **R** **D**;

3- Pós-ordem: **E** **D** **R**;

2- A-B-**C**



# Percurso em Árvore Binária

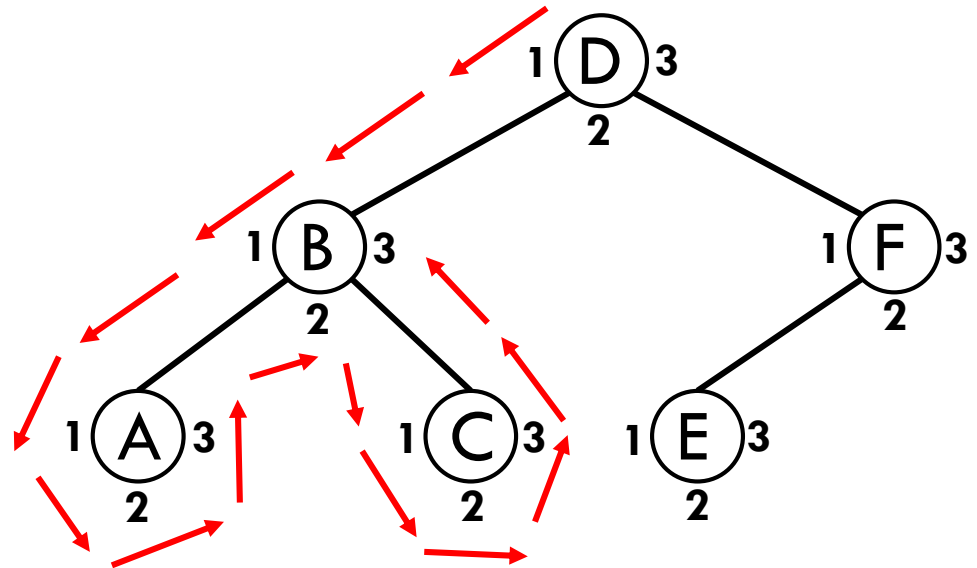
24

1- Pré-ordem: **R** **E** **D**;

2- Em-ordem: **E** **R** **D**;

3- Pós-ordem: **E** **D** **R**;

2- A-B-C





# Percurso em Árvore Binária

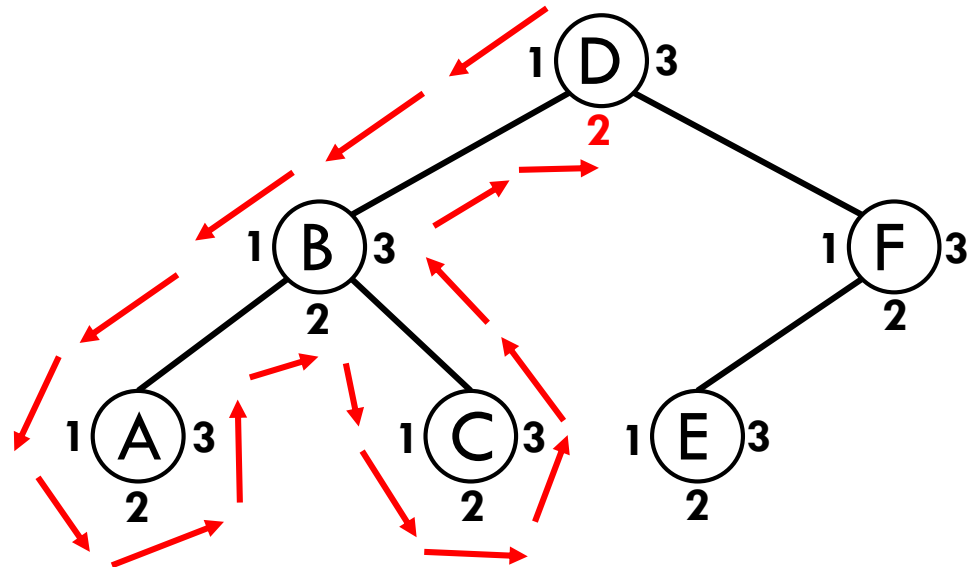
25

1- Pré-ordem: **R** E D;

2- Em-ordem: E **R** D;

3- Pós-ordem: E D **R**;

2- A-B-C-D



# Percurso em Árvore Binária

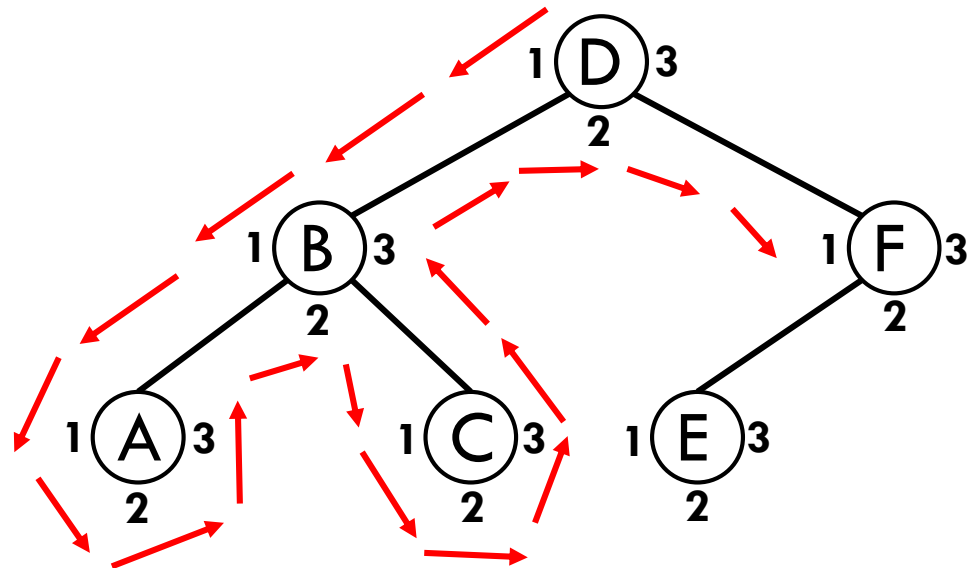
26

1- Pré-ordem: **R** **E** **D**;

2- Em-ordem: **E** **R** **D**;

3- Pós-ordem: **E** **D** **R**;

2- A-B-C-D



# Percurso em Árvore Binária

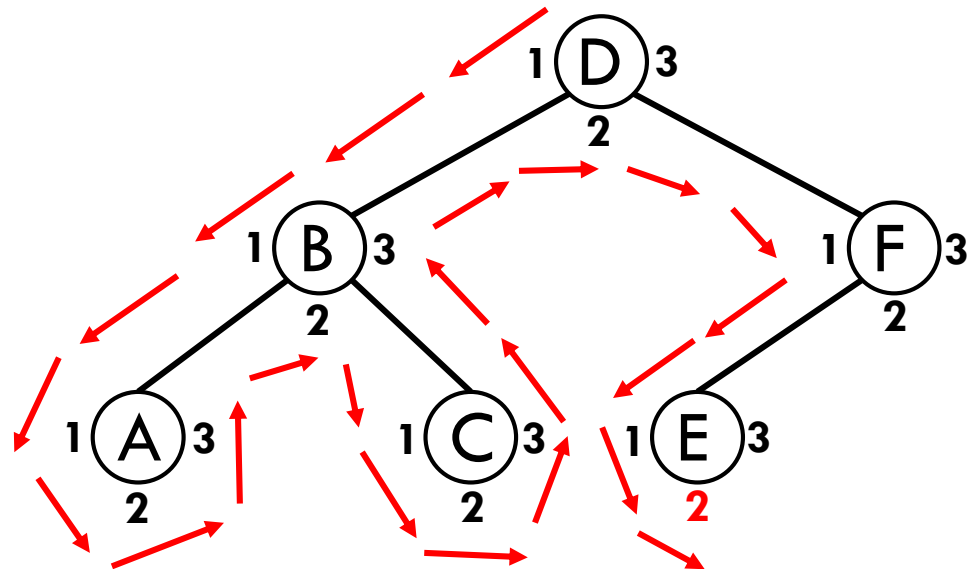
27

1- Pré-ordem: **R** **E** **D**;

2- Em-ordem: **E** **R** **D**;

3- Pós-ordem: **E** **D** **R**;

2- A-B-C-D-**E**



# Percurso em Árvore Binária

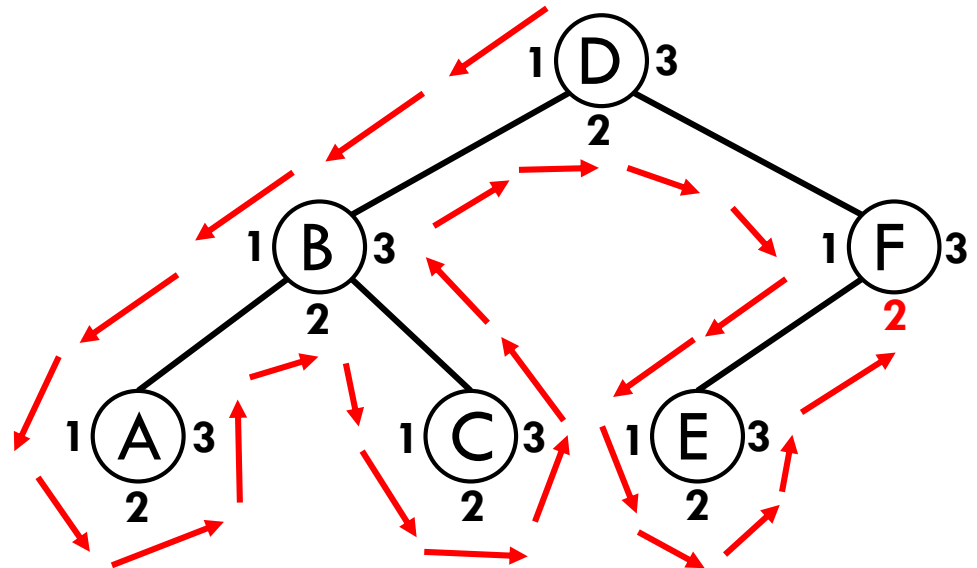
28

1- Pré-ordem: **R** **E** **D**;

2- Em-ordem: **E** **R** **D**;

3- Pós-ordem: **E** **D** **R**;

2- A-B-C-D-E-**F**



# Percurso em Árvore Binária

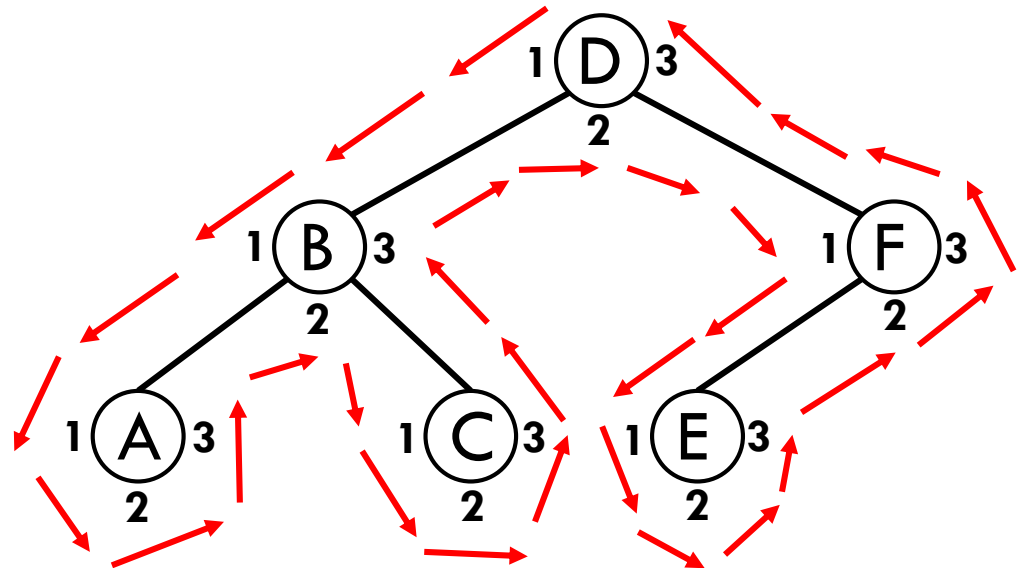
29

1- Pré-ordem: **R** **E** **D**;

2- Em-ordem: **E** **R** **D**;

3- Pós-ordem: **E** **D** **R**;

2- A-B-C-D-E-F



# Árvore binária

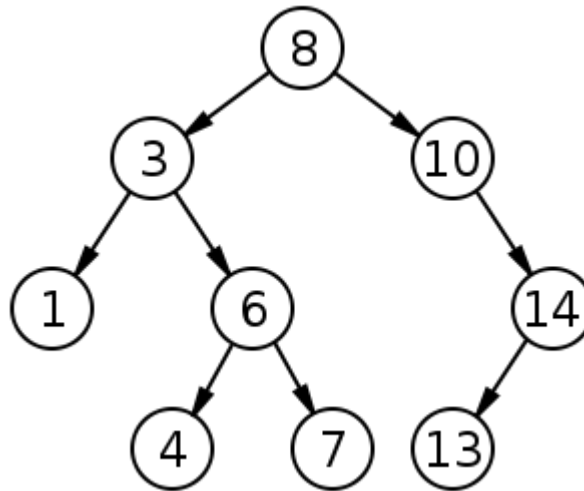
30

- Uma árvore binária pode ser implementada **estaticamente** em um vetor.
- São utilizadas duas funções para determinar os índices dos nós:
  - ▣  $\text{Esquerda}(\text{pai}) = 2 * \text{pai} + 1$
  - ▣  $\text{Direita}(\text{pai}) = 2 * \text{pai} + 2$
- Tamanho do vetor:  $2^h - 1$ 
  - ▣ Pior caso, árvore binária completa.

# Árvore binária

31

- $\text{Esquerda}(\text{pai}) = 2 * \text{pai} + 1$
- $\text{Direita}(\text{pai}) = 2 * \text{pai} + 2$

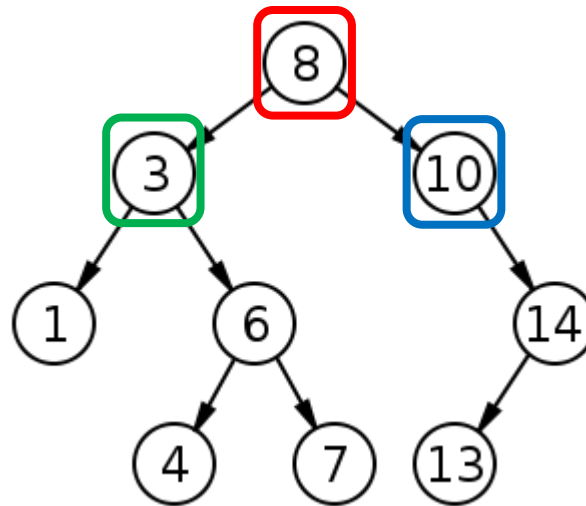


0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
8	3	10	1	6	-	14	-	-	4	7	-	-	13	-

# Árvore binária

32

- $\text{Esquerda}(\text{pai}) = 2 * \text{pai} + 1$
- $\text{Direita}(\text{pai}) = 2 * \text{pai} + 2$



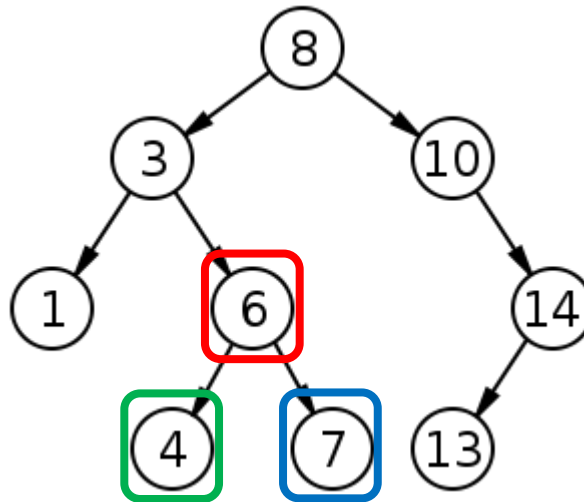
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
8	3	10	1	6	-	14	-	-	4	7	-	-	13	-



# Árvore binária

33

- $\text{Esquerda}(\text{pai}) = 2 * \text{pai} + 1$
- $\text{Direita}(\text{pai}) = 2 * \text{pai} + 2$

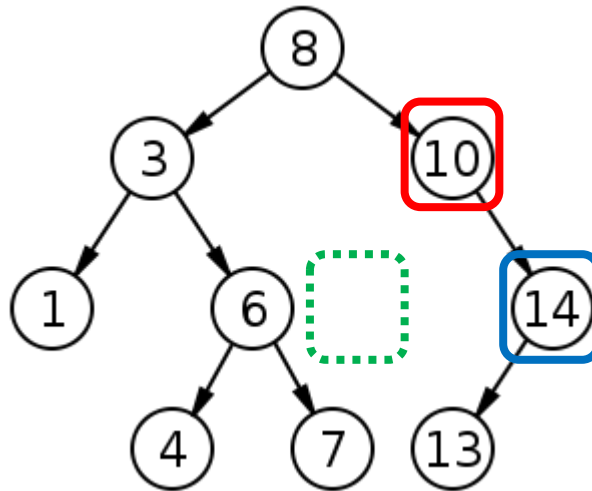


0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
8	3	10	1	6	-	14	-	-	4	7	-	-	13	-

# Árvore binária

34

- $\text{Esquerda}(\text{pai}) = 2 * \text{pai} + 1$
- $\text{Direita}(\text{pai}) = 2 * \text{pai} + 2$



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
8	3	10	1	6	-	14	-	-	4	7	-	-	13	-

# Árvore binária

35

- A implementação dinâmica de uma AB consiste no encadeamento de vários nós.
- Cada nó é representado por: 

*Esq	Dado	*Dir
------	------	------

