



# Estruturas Avançadas de Dados I

## Árvores Binárias

1

## Definição I

- Uma árvore binária  $T$  é um conjunto finito de elementos denominados nós ou vértices, tal que:
  - Se  $T = \emptyset$ , a árvore é dita vazia, ou
  - Existe um nó especial  $r$ , chamado raiz de  $T$ , e os restantes podem ser divididos em dois subconjuntos disjuntos,  $Tr^E$  e  $Tr^D$ , a sub-árvore esquerda e a direita de  $r$ , respectivamente, as quais são também árvores binárias.
  - A raiz da sub-árvore esquerda (direita) de um nó  $v$ , se existir, é denominada filho esquerdo (direito) de  $v$ . Naturalmente, o esquerdo pode existir sem o direito e vice-versa. Se  $r$  é a raiz de  $T$ , diz-se, também, que  $Tr^E$  e  $Tr^D$  são as sub-árvores esquerda e direita de  $T$ , respectivamente.
  - Uma árvore binária pode ter duas sub-árvores vazias (a esquerda e a direita). Toda árvore binária com  $n$  nós possui exatamente  $n + 1$  sub-árvores vazias entre suas sub-árvores esquerdas e direitas.



2

## Definição II

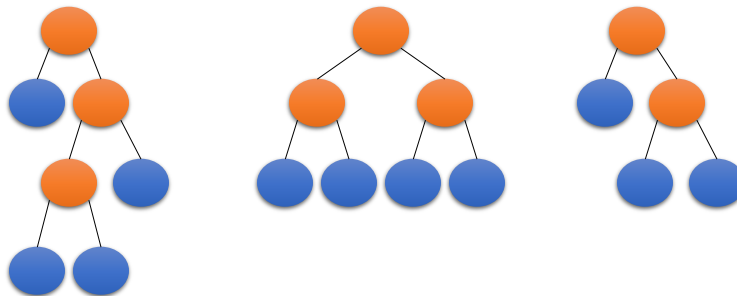
- Uma árvore binária é uma árvore cujos os nós têm 0, 1 ou 2 filhos e cada filho é designado como filho à esquerda ou filho à direita (**Grau 2**);
- O número de folhas é uma importante característica das árvores binárias para mensurar a eficiência esperada de algoritmos.



3

## Classificação de Árvores Binárias

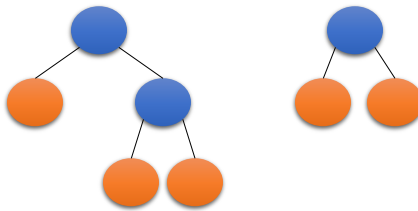
- Árvore Estritamente Binária
  - Se cada **nó não terminal** em uma árvore binária ter sub-árvores à esquerda e direita, ou nenhum nó tem **filho único**;
  - Uma árvore com  $n$  folhas, terá  $2n - 1$  nós.



4

## Classificação de Árvores Binárias

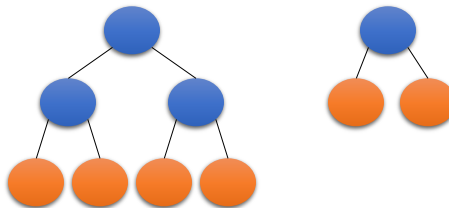
- Árvore Binária Completa (ou Quase Completa)
  - Uma árvore binária de nível  $n$  é uma árvore binária completa se:
    - cada nó com menos de dois filhos deve estar no nível  $n$  ou no nível  $n-1$



5

## Classificação de Árvores Binárias

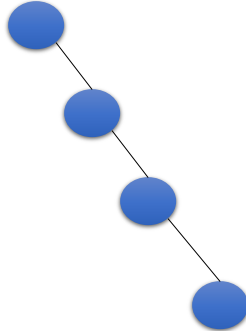
- Árvore Binária Cheia
  - Uma árvore binária cheia de nível  $n$  é a árvore estritamente binária, onde todos os nós folhas estão no nível  $n$ ;
  - Uma árvore cheia com altura  $h$ , terá  $2^h - 1$  nós.



6

## Classificação de Árvores Binárias

- Árvore Degenerada
  - Cada nó possui exatamente um filho, e a árvore tem o mesmo número de níveis que de nós.



## Percursos (caminhamento) em Árvores Binárias

- Conforme mencionado na nota de aula anterior, temos dois percursos:
  - Percurso em profundidade: os nós da sub-árvore atual têm prioridade na ordem de acesso;
  - Percurso em largura (amplitude): os nós de menor nível têm prioridade na ordem de acesso.

## Percursos (caminhamento) em Árvores Binárias

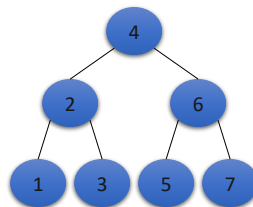
- Em profundidade, temos três tipos “canônicos”:
  - Pré-ordem
  - Pós-ordem
  - Em-ordem
- Em amplitude, temos:
  - Em nível



9

## Caminhamento Pré-ordem

- Visitar a raiz
- Percorrer a sub-árvore esquerda em pré-ordem
- Percorrer a sub-árvore direita em pré-ordem



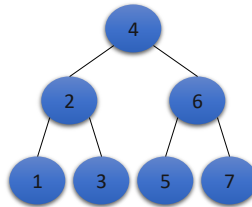
Percurso: 4, 2, 1, 3, 6, 5, 7



10

## Caminhamento Pós-ordem

- Percorrer a sub-árvore esquerda em pós-ordem
- Percorrer a sub-árvore direita em pós-ordem
- Visitar a raiz



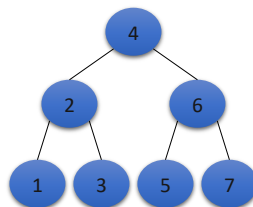
Percurso: 1, 3, 2, 5, 7, 6, 4



11

## Caminhamento Em-ordem

- Percorrer a sub-árvore esquerda em Em-ordem
- Visitar a raiz
- Percorrer a sub-árvore direita em Em-ordem



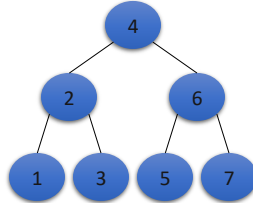
Percurso: 1, 2, 3, 4, 5, 6, 7



12

## Caminhamento Em Nível

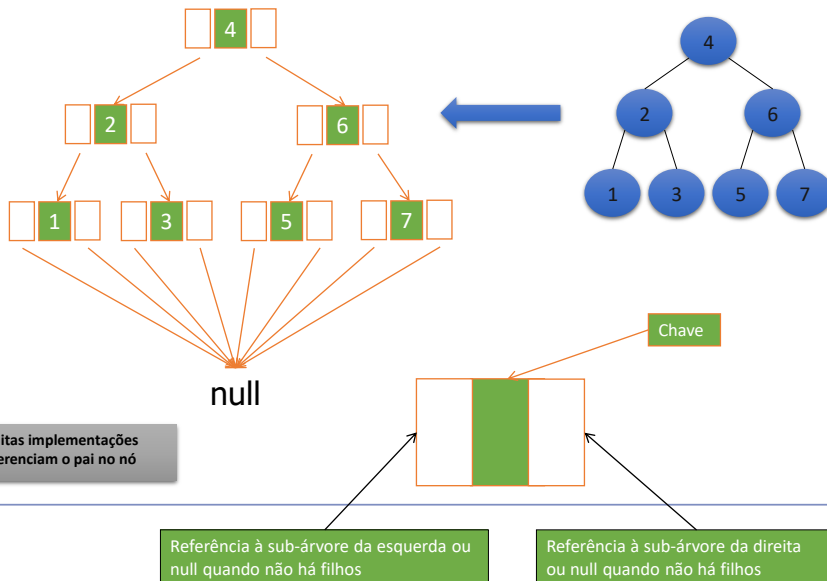
- Percorre-se a árvore em nível de cima para baixo e da esquerda para a direita.



Percurso: 4, 2, 6, 1, 3, 5, 7

13

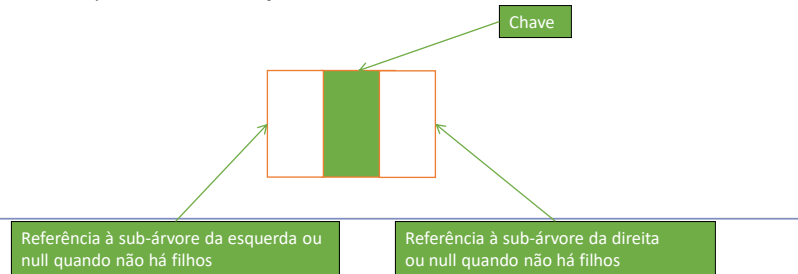
## Estrutura de dados de uma Árvore Binária



14

# Árvore Binária de Pesquisa

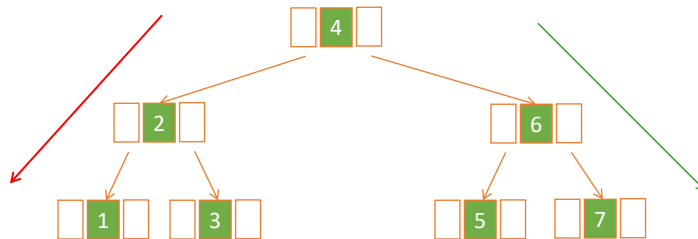
- Também chamada de:
  - árvore binária de busca
  - ou árvore binária ordenada
- Apresentam uma relação de ordem entre os nós;
- Ordem é definida por um campo denominado **chave**;
- Não permite chave duplicada, ou seja, cada nó tem um valor de chave diferente.



17

# Árvore Binária de Pesquisa

Exemplo



**Esquerda:**  
Valor da chave **inferior** ao  
valor da chave do nó pai.

**Direita:**  
Valor da chave **superior** ao  
valor da chave do nó pai.

18



## Árvore Binária de Pesquisa – Buscar Valor

- A procura de um valor em uma árvore binária é algo mais rápido do que a procura em listas encadeadas ou arrays;
- Para cada nó, compare a chave a ser localizada com o valor armazenado no nó atual;
- Se a chave for menor, vá para a sub-árvore esquerda, senão vá para a sub-árvore direita e faça o procedimento anterior;
- A busca para quando for encontrado o nó ou quando não há mais meios de continuar (nó folha), pois a chave não está na árvore.



19

## Árvore Binária de Pesquisa – Buscar Valor

- A complexidade pode ser medida pelo número de comparações feitas durante o processo de busca;
- Dependerá do número de nós encontrados no único caminho que leva da raiz ao nó procurado;
- Então a complexidade, depende da forma da árvore e da posição do nó procurado na árvore. O número médio de comparações em uma busca é  $O(\log(n))$ , pois a altura de uma árvore binária completa é  $h = \log(n+1)$ .



20

## Árvore Binária de Pesquisa – Exclusão

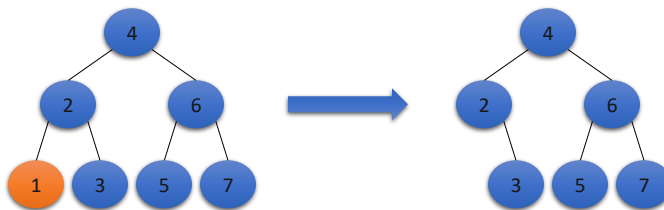
- Existem três casos ao excluir um nodo:
  - Exclusão de uma folha;
  - Exclusão de um nó que possui UM filho;
  - Exclusão de um nó que possui DOIS filhos;
    - Exclusão por cópia;
    - Exclusão por fusão;



24

## Árvore Binária de Pesquisa – Exclusão

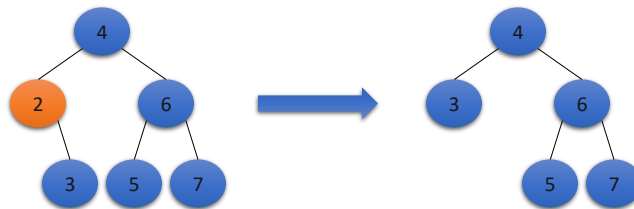
- Caso 1: Exclusão de uma folha
  - Na referência do nodo pai é atribuído o valor null.



25

## Árvore Binária de Pesquisa – Exclusão

- Caso 2: Exclusão de uma nó que possui UM filho
  - A referência do nó pai aponta para o nó filho do nó excluído.

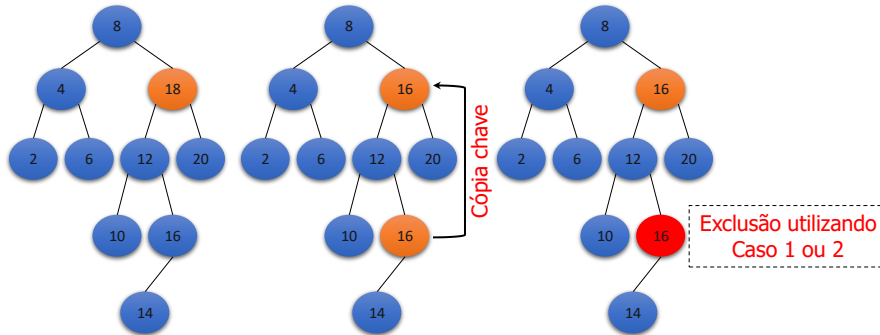


## Árvore Binária de Pesquisa – Exclusão por Cópia

- Exclusão por Cópia (um nó que possui DOIS filhos)
  - Remove uma chave k1 (chave do nó à excluir):
    - Sobrescrevendo-a por uma outra chave k2 (o maior valor na sub-árvore esquerda);
    - Então removendo o nó que contém k2 (que será um dos casos simples: folha, ou nó com apenas um filho).

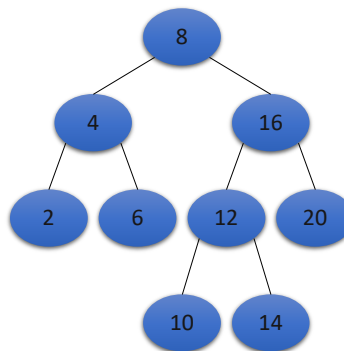
## Árvore Binária de Pesquisa – Exclusão por Cópia

- Exclusão por Cópia (um nó que possui DOIS filhos)



## Árvore Binária de Pesquisa – Exclusão por Cópia

- Exclusão por Cópia (um nó que possui DOIS filhos)



Árvore após exclusão por cópia

## Árvore Binária de Pesquisa – Exclusão por Fusão

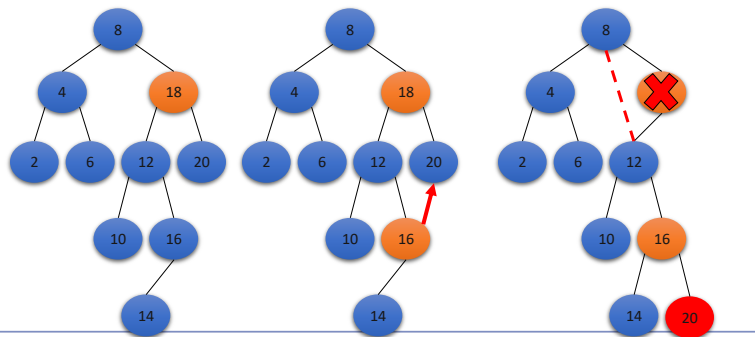
- Exclusão por Fusão (um nó que possui DOIS filhos)
  - A solução consiste em fusionar as duas sub-árvores do nó a ser excluído em uma;
  - Para tanto, como na organização da árvore binária, todos os valores da sub-árvore à esquerda são menores que os valores da sub-árvore à direita, deve-se encontrar o maior valor na sub-árvore esquerda e torná-lo a raiz da sub-árvore direita. Também pode-se procurar o nó com menor valor da sub-árvore direita;
  - Remove a chave, excluindo o nó que contém a chave. E o pai do nó removido passa a apontar para a nova sub-árvore.



32

## Árvore Binária de Pesquisa – Exclusão por Fusão

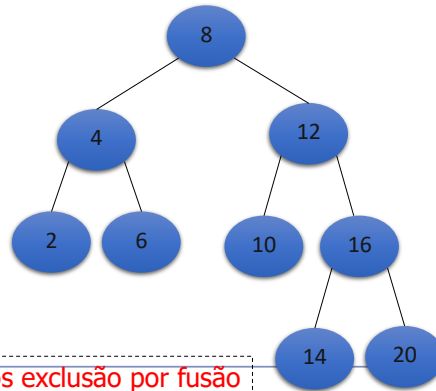
- Exclusão por Fusão (um nó que possui DOIS filhos)



33

## Árvore Binária de Pesquisa – Exclusão por Fusão

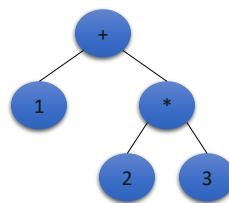
- Exclusão por Fusão (um nó que possui DOIS filhos)



Árvore após exclusão por fusão

34

## Árvores de Expressão

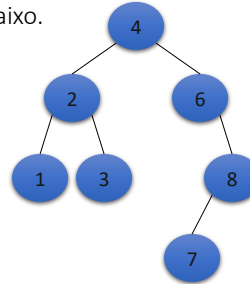


- A árvore de expressão acima representa “1 + 2 \* 3”
- Os operandos sempre estarão nos nós folhas e, operadores nos nós não terminais;
- Pós-ordem: 1 2 3 \* + (notação polonesa)
- Em-ordem: 1 + 2 \* 3 (ordem da expressão)
- Pré-ordem: + 1 \* 2 3

36

## Exercícios

- 3.1 - Criar uma árvore binária de pesquisa inserindo as chaves conforme a ordem: **g, d, a, c, m, o, b, d, a, l, p, q**
- 3.2 – Desenhe uma árvore de pesquisa com a inserção dos elementos na seguinte ordem: K G G B L O C P A U D  
Apresente os quatro percursos estudados!
- 3.3 – Desenhe uma árvore de pesquisa com a inserção dos elementos na seguinte ordem: 20 10 5 8 12 3 14 23 28 30 100 e 99
- 3.4 – Apresente o percurso pós-ordem para a árvore abaixo.

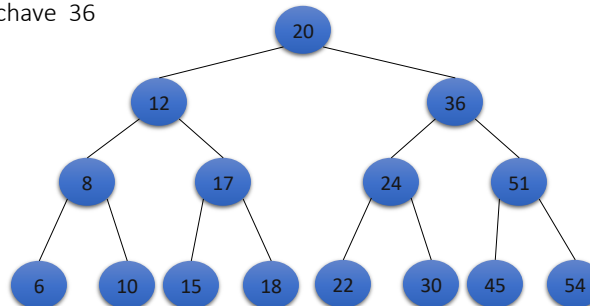


37

## Exercícios

3.5 – A partir da árvore abaixo, aplique as seguintes técnicas de exclusão:

- Exclusão por fusão da chave 12
- Exclusão por cópia da chave 36
- Exclusão da chave 8
- Exclusão da chave 6

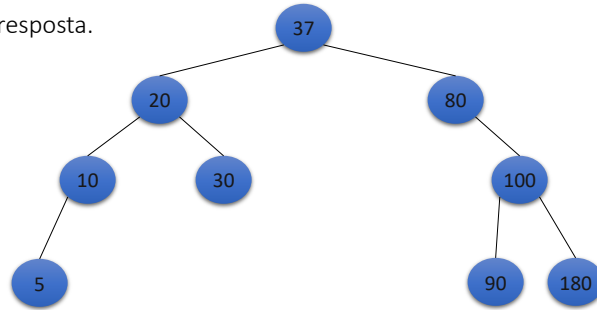


38

## Exercícios

3.6 – A árvore abaixo é:

- Estritamente binária? Justifique sua resposta.
- Completa? Porque? Justifique sua resposta.
- Cheia? Porque? Justifique sua resposta.



39

## Exercícios

3.7 – Desenhe a árvore de expressão para o seguinte cálculo:

$$3 * 5 + 8 / (10 - 3)$$



40



## Referências Bibliográficas

- ASCENCIO, A. F. G; ARAÚJO, G. S. Estruturas de Dados. São Paulo: Pearson Prentice Hall, 2010. 432 p.
- BATISTA, C. A. T. Estruturas de Dados. Lâminas segundo semestre, 2009.
- CORMEN, Thomas H. et al. Introduction to algorithms. 3. ed. Cambridge: MIT, 2009. xix. 1292 p.
- JAQUES, Patrícia. Lâminas Árvores Binárias – Programação II, Unisinos.
- SZWARCFITER, J.; MARKENZON, L. Estruturas de Dados e seus Algoritmos. Rio de Janeiro: LTC, 1994.



**Prof. Márcio Garcia Martins**  
**marciog@unisinos.br**

Para anotar: *ao enviar e-mail sempre coloque o  
 seguinte prefixo no assunto*  
**[EADI-ano-semester] – Nome do aluno**

