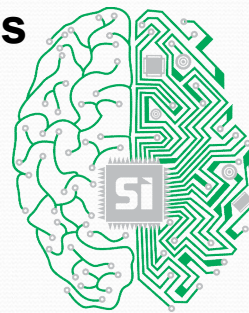


Estrutura de Dados

Árvores Binárias

Prof. Fernando Del Moro



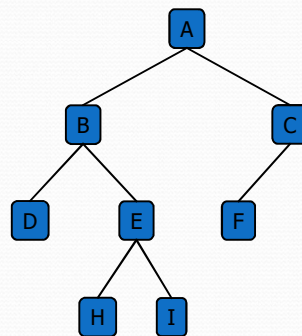
SISTEMAS DE INFORMAÇÃO 1

Árvore Binária

- Uma **árvore binária** ou **BT** (*binary tree*) é uma estrutura de dados caracterizada por:
- Ou não tem elemento algum (árvore vazia).
- Ou tem um elemento Inicial, denominado **raiz**, com **dois ponteiros** (daí o nome binária) para duas estruturas diferentes, denominadas **sub-árvore esquerda** e **sub-árvore direita**.

Árvore Binária

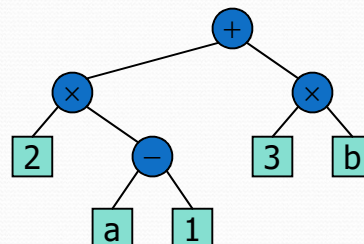
- Aplicações:
 - Expressões aritméticas
 - Processo de decisão
 - Busca
 - Ordenação



3

Árvore de Expressões Aritméticas

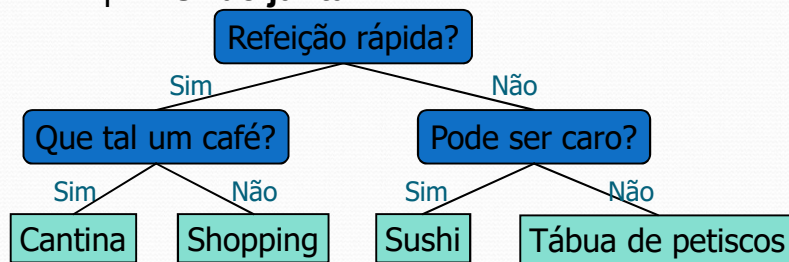
- Árvore binária associada com uma expressão aritmética
 - Nós internos: operadores
 - Nós externos: operandos
- Exemplo: árvore da expressão aritmética para a expressão $(2 \times (a - 1) + (3 \times b))$



4

Árvore de decisão

- Árvore binária associada com um processo de decisão
 - Nós internos: questões com respostas sim/não
 - Nós externos: decisões
- Exemplo: **Onde jantar?**



5

Propriedades de AB (BT)

- Notação

n número de nós

e número de nós externos

i número de nós internos

h altura (height)

- Propriedades:

– $e = i + 1$

– $n = 2e - 1$

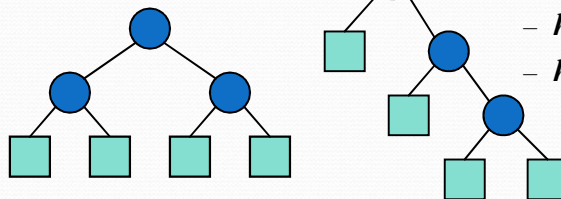
– $h \leq i$

– $h \leq (n - 1)/2$

– $e \leq 2^h$

– $h \geq \log_2 e$

– $h \geq \log_2 (n + 1) - 1$



6

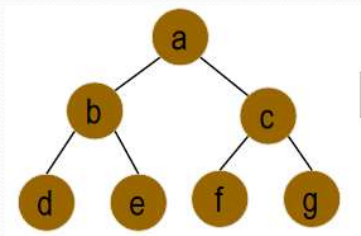
Definições para Árvores Binárias

- Os nós de uma árvore binária possuem **graus** zero, um ou dois.
- Um nó de grau zero é denominado **folha**.
- Um nó de grau dois ou um é denominado **interno** ou **interior**.
- Uma árvore **estritamente binária** é uma árvore onde todos os nós tem grau **zero** ou **dois**.

7

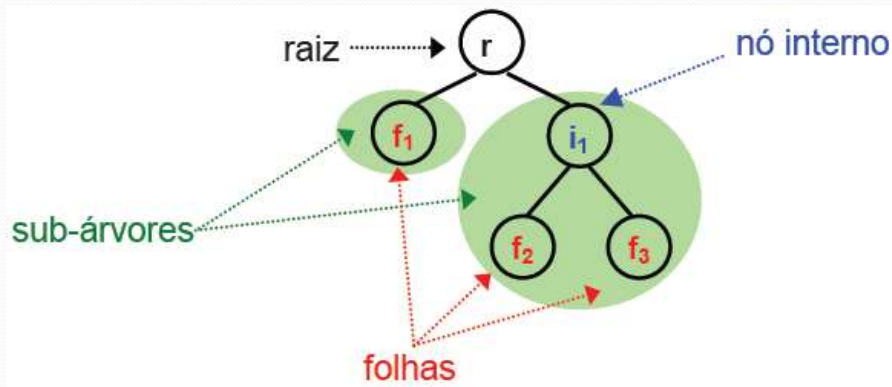
Definições para Árvores Binárias

- O número de arestas de um caminho define o COMPRIMENTO do caminho.
- Arvore Binária Completa(ABC)



8

Definições para Árvores Binárias



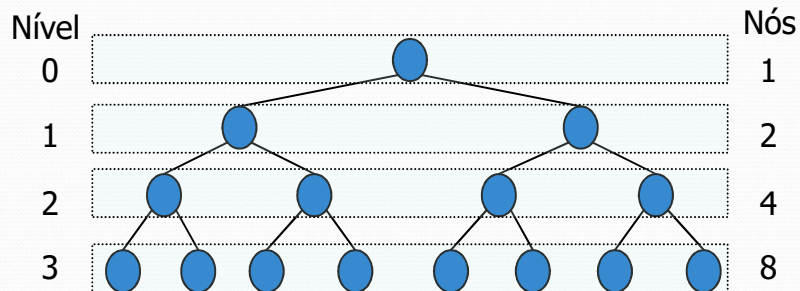
9

Definições para árvores binárias

- A **profundidade** de um nó é a distância deste nó até a raiz.
- Um conjunto de nós com a mesma profundidade é denominado **nível** da árvore.
- A **altura** de um nó "x" em uma árvore binária é a distância entre "x" e o seu descendente mais afastado.
- A **altura** do nó **Raiz** é a altura da **Árvore**.

10

Propriedades de AB (BT)



Número máximo de nós em um nível h é 2^h

Número total de nós é, no máximo é $2^{h+1}-1$

11

Percursos em árvores binárias

- Uma operação muito comum é **percorrer** uma árvore binária, o que significa passar por todos os nós, pelo menos uma vez.
- O conceito de **visitar** significa executar uma operação com a informação armazenada no nó (imprimir, testar, etc.)
- Na operação de percorrer a árvore cada nodo é acessado 3 vezes, **porém existe apenas 1 visita.**

12

Percursos em árvores binárias

- Não existe ordem natural para percorrer árvores e portanto podemos escolher diferentes maneiras de percorrê-las.
- Nós iremos estudar três métodos para percorrer árvores.
- Todos estes três métodos se baseiam em três operações básicas:
 - Visitar a raiz,
 - Percorrer a subárvore da esquerda e
 - Percorrer a subárvore da direita.

13

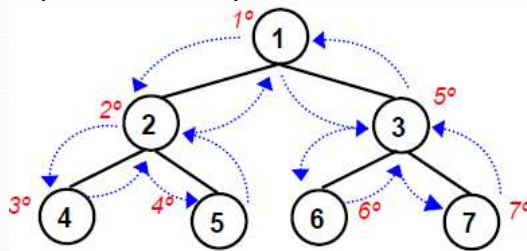
Percursos em árvores binárias

- A única diferença entre estes métodos é a hora em que a visita ocorre.
- Há três maneiras recursivas de se percorrer árvores binárias:
 - Percurso em **pré-ordem**
 - Percurso em **pós-ordem**
 - Percurso em **in-ordem**

14

Percurso pré ordem

- se árvore vazia; fim
- visitar o nó raiz
- percorrer em pré-ordem a subárvore esquerda
- percorrer em pré-ordem a subárvore direita

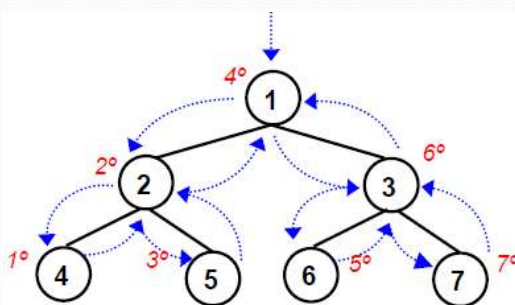


- **Resultado:**
- 1 2 4 5 3 6 7
- Dica: visita o nó quando passar a sua esquerda

15

Percurso in-ordem

- Se árvore vazia, fim
- Percorrer em in-ordem a subárvore esquerda
- Visitar o nó raiz
- Percorrer em in-ordem a subárvore direita

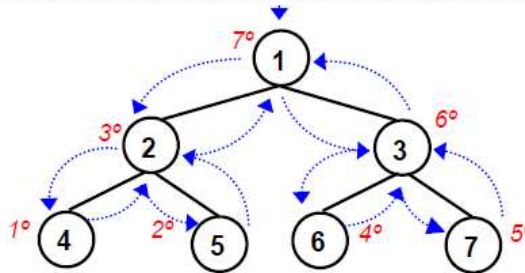


- **Resultado:**
- 4 2 5 1 6 3 7
- Dica: Visita o nó quando passar embaixo do nó

16

Percurso pos-ordem

- Se árvore vazia, fim
- Percorrer em Pós-Ordem a subárvore esquerda
- Percorrer em Pós-Ordem a subárvore direita
- Visitar o nó raiz

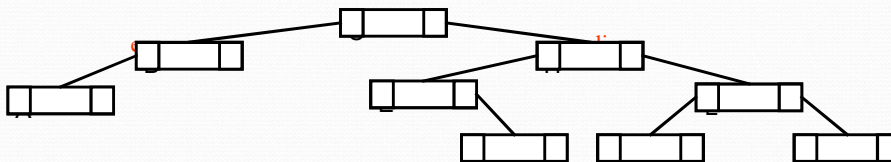


- **Resultado:**
- 4 5 2 6 7 3 1
- Dica: Visita o nó quando passar a sua direita

17

E se os nós da árvore estivessem ordenados?

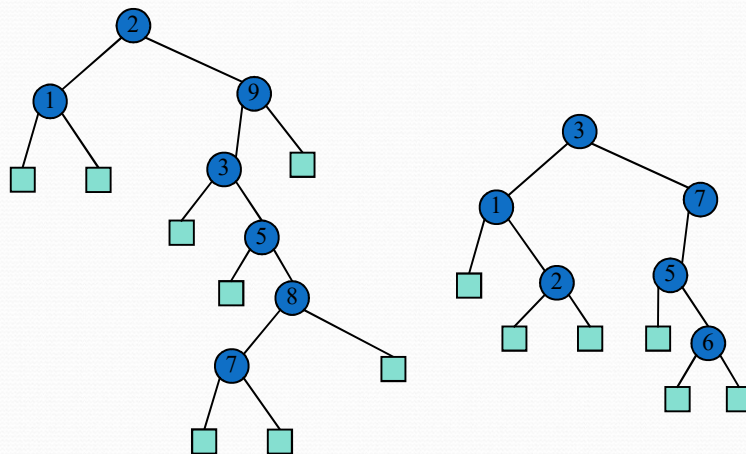
raiz



- Conhecida como Árvore de Pesquisa Binária:
- Nós da sub-árvore esquerda são menores do que a raiz
- Nós da sub-árvore direita são maiores do que a raiz

18

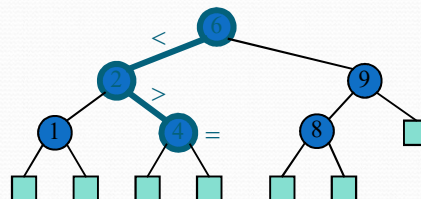
Árvore de pesquisa binária



19

Busca

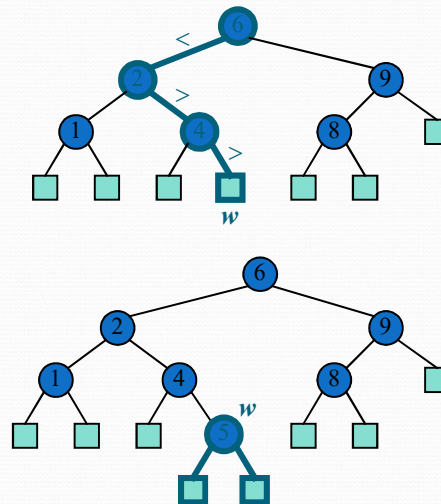
- Para procurar uma chave k , procuramos a partir da raiz comparando com a chave do nó.
- O próximo nó depende da comparação da chave pesquisada com a chave do nó atual
- Se chegar em uma folha e não encontrar a chave, a busca não obteve sucesso



20

inserção

- Para executar $\text{inserção}(k, o)$, procura-se pela chave k
- Assumindo que k ainda não está na árvore, w será a folha encontrada pela busca
- Inserimos k no nó w
- Exemplo: inserir 5



21

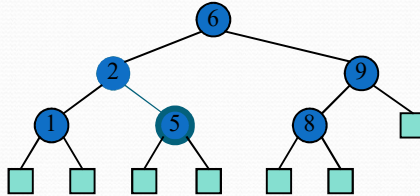
remoção

- Para remover um nó de uma árvore binária devemos considerar três casos:
 - nó sem filhos;
 - nó com um único filho;
 - nó com dois filhos.

22

remoção

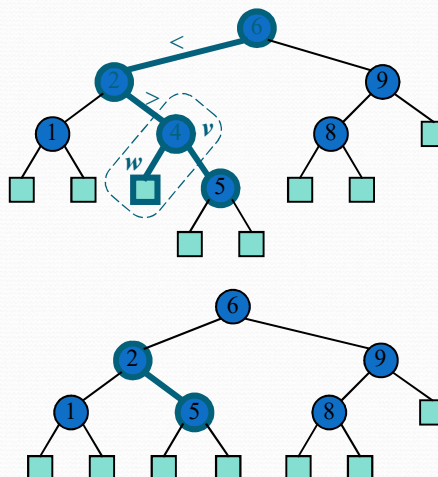
- Para executar `remove(k)`, procuramos pela chave k
- *Se o nó for folha, ou seja, um nó sem filhos, é o mais simples e basta apenas ajustar o ponteiro de seu pai*



23

remoção

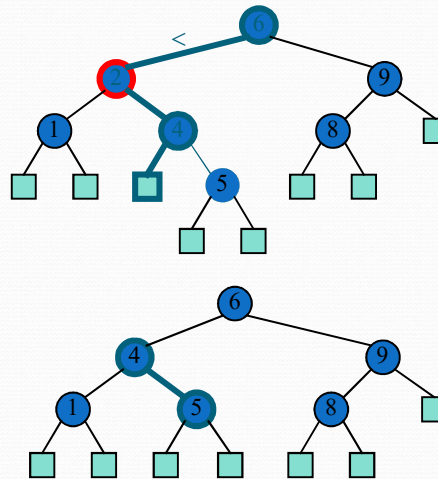
- *No caso do nó ter um único filho a mudança na árvore também é simples e basta mover o nó filho daquele que será removido uma posição para cima, ou seja, remova-o e o seu filho assume o seu lugar*



24

remoção

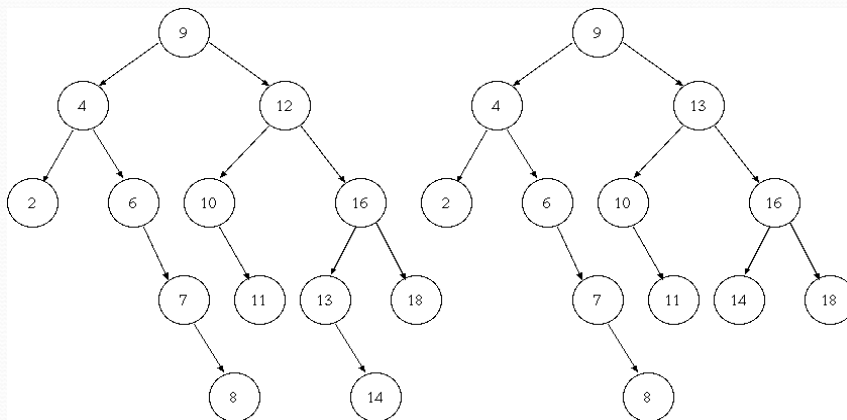
- Se o nó tiver dois filhos, encontre o sucessor através de um caminhamento até o item mais a esquerda da sub-árvore direita do nó a ser removido.
- O nó sucessor assume o lugar do nó removido



25

remoção

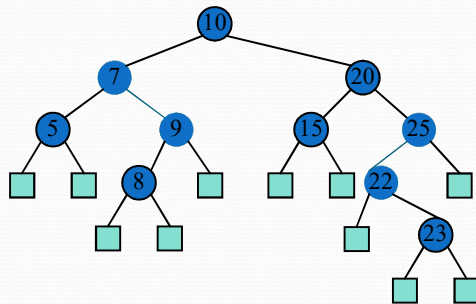
- Outro exemplo de remoção. No exemplo, o valor 12 será removido



26

Exemplo de remoção

- *Remover 5, 9 e 20*



27

Definição da Estrutura de Dados Árvore Binária usando ponteiros

- ```
class Nodo{
 private String informacao;
 private Nodo esquerda;
 private Nodo direita;
}
```
- ```
class Arvore{  
    private Nodo raiz;  
}
```

28