

## Aplicações de árvores binárias

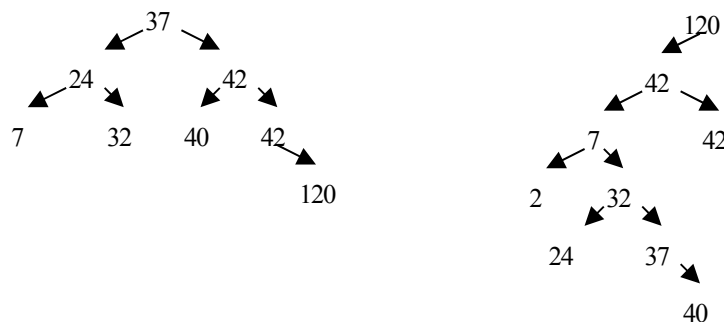
Há várias aplicações que precisam de estruturas representando hierarquia ou composição para armazenamento dos dados manipulados. Algumas dessas aplicações utilizam:

- **Árvores Binárias de Pesquisa (ABP)**
- **Representações de expressões aritméticas**
- **Outros:** representação do espaço bidimensional (*quadtrees*) e tridimensional (*octrees*), representação de objetos compostos, árvores de decisão.

### 1) Árvores Binárias de Pesquisa

Árvore binária que armazena símbolos em função de uma relação de ordem. Assim, uma ABP tem como propriedade básica:

- todos os nodos armazenados na sub-árvore a esquerda de um nodo cujo valor é  $k$  tem valor menor que  $k$ . Todos os nodos da sub-árvore da direita tem valor maior ou igual a  $k$ .



**Propriedade fundamental:** todos os elementos armazenados na sub-árvore à esquerda de um nodo  $k$  serão menores que  $k$ , e os elementos armazenados na sub-árvore à direita serão maiores que  $k$ . Ou vice-versa.

Operações básicas:

- Inserir(símbolo, árvore)
- Existe(símbolo, árvore)
- Remover(símbolo, árvore)

**2.1. Operação de inserção:** símbolo inserido é sempre folha.

Regras para inserção dos elementos:

- Se árvore é vazia, o símbolo é colocado na raiz;
  - Se o símbolo é *menor que a raiz*, ele é colocado na *sub-árvore da esquerda*;
  - Se o símbolo é *maior do que a raiz*, ele é colocado na *sub-árvore da direita*.
-

Exercício: inserir 10 – 8 – 11 – 5 – 2 – 9 – 15 – 7 – 6 – 12 – 20 em uma árvore binária de pesquisa.

---

**Característica importante das ABP:** Quando enumerados através de um processo de pesquisa “em ordem” ou “central”, os valores armazenados em uma ABP são exibidos em uma lista, do menor para o maior se os menores foram sempre inseridos à esquerda.

**2.2. Operação de “existe”:** pesquisa para verificar se um símbolo pertence à árvore.

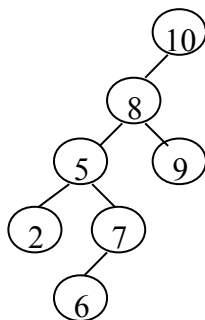
Regras:

- se símbolo for igual à raiz, retorna símbolo ou V;
- se símbolo for menor que a raiz, realizar **Existe** na sub-árvore da esquerda; se for maior, **Existe** na sub-árvore da direita.

**2.3. Operação de remoção:** remoção de um símbolo, mantendo os demais ordenados.

Regras:

- se o nodo a ser removido não possui sub-árvores, remove o nodo;
- se o nodo a ser removido possui ou sub-árvore à esquerda ou sub-árvore à direita, esta sub-árvore passa a ocupar o seu lugar e ele é removido;
- se o nodo a ser removido possui duas sub-árvores: coloca no seu lugar ou o maior nodo da sua sub-árvore à esquerda ou o menor nodo da sua sub-árvore à direita



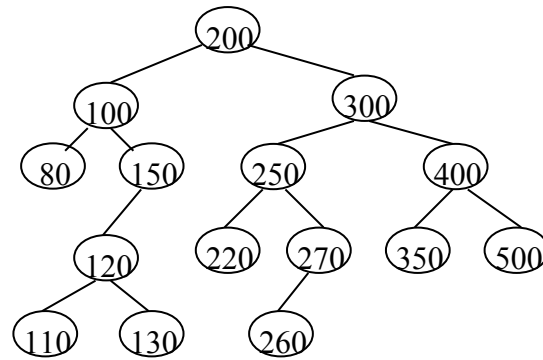
Por exemplo, para remover 8:

- maior nodo da sub-árvore à esquerda = 7
- menor nodo da sub-árvore à direita = 9

***Para achar o maior valor à esquerda de um nodo:*** ir para a sua sub-árvore da esquerda, e à partir daí ir sempre para o nodo à direita, até encontrar próximo=nulo.

***Para achar o menor valor à direita de um nodo:*** ir para a sua sub-árvore da direita, e à partir daí ir sempre para o nodo à esquerda, até encontrar próximo=nulo.

**Exercícios:** dada a seguinte árvore binária de pesquisa:



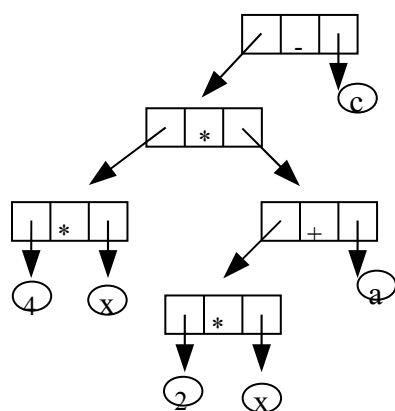
1. Redesenhar a árvore após remover o 150;
  2. Redesenhar a árvore (original) após remover 300, considerando:
    - a) promoção do menor valor à direita:
    - b) promoção do maior valor à esquerda:
  3. Escreva a sequência de nodos gerada pelos caminhamentos central, pré e pós-fixado (à esquerda e à direita), sobre a árvore original.
-

## 2) Representação de expressões aritméticas

- Operandos são sempre folhas das árvores
- Operadores são sempre raízes das sub-árvores
- Operador de menor prioridade aparece na raiz da árvore; sub-expressão à esquerda deste operador aparece na sub-árvore da esquerda; sub-expressão à direita aparece na sub-árvore da direita.

Árvore binária representando uma expressão aritmética composta de operadores binários.

Exemplo:  $4x(2x + a) - c$



Exercícios: representar sob forma de árvore binária:

1)  $5+3*4$

2)  $(5+3)*4$

3)  $(3+4)*5+2+7*8$

4)  $x := 5/a+(b-1)$

**Caminhamentos em expressões representadas por árvores:** definem a sequência de entrada dos dados para o cálculo, ou a sequência usada para avaliação da expressão:

a) **Pré-fixado:** expressão em *notação polonesa*.

Exemplo: 1)  $+5*34$

2)  $*+534$

- operadores antes dos operandos

- b) **Pós-fixado**: expressão em *notação polonesa reversa* (usada em máquinas de calcular HP).

Exemplo: 1)  $534*+$

2)  $53+4*$

- *operandos antes dos operadores*

- c) **Central**: expressão na forma original.

Exemplo: 1)  $5+3*4$

2)  $5+3*4$

---

Exercícios:

1. Faça um algoritmo que leia uma expressão armazenada em uma árvore binária, escrevendo-a na sua forma normal totalmente parentizada (considere que cada sub-expressão é composta de dois operandos e um operador). Exemplo:  $(5+(3*4))$  ou  $((5+3)*4)$ .
2. Dadas as seguintes expressões, monte a respectiva representação em árvore binária e escreva a expressão de origem:

a) polonesa:  $/ + 10\ 20,5\ * + 3\ 4\ 2$       b) polonesa reversa:  $5\ 3\ 0,5\ /\ * 4\ 3\ 9 - 2 /\ * +$