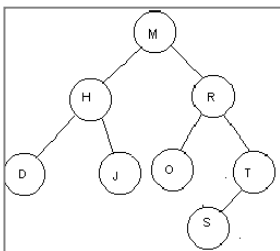


TP10 – Lista de Exercícios P2

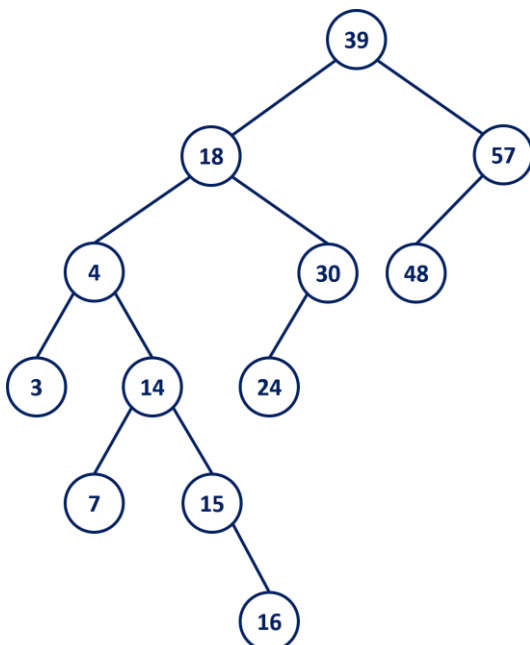
- Recursividade,
- Árvore Binária - conceitos
- Árvore Binária de Busca – Conceitos (inserção, remoção, busca, máx., min),
- Lista Ordenada
- Tabela Hashing - conceitos

1) Dada a árvore, indique:



- Os nós folha: **D, J, O, S**
- Grau da árvore: **2**
- Altura da árvore: **4**
- Os filhos de R: **O, T**

2) Dada uma árvore de Busca Binária (BST) inicialmente vazia, insira e desenhe os seguintes elementos e nessa ordem: 39, 18, 57, 4, 14, 30, 48, 7, 3, 15, 24, 16. Após, escreva os percursos:



- Pré-ordem

39, 18, 4, 3, 14, 7, 15, 16, 30, 24, 57, 48

- Em ordem

3, 4, 7, 14, 15, 16, 18, 24, 30, 39, 48, 57

- Pós-ordem

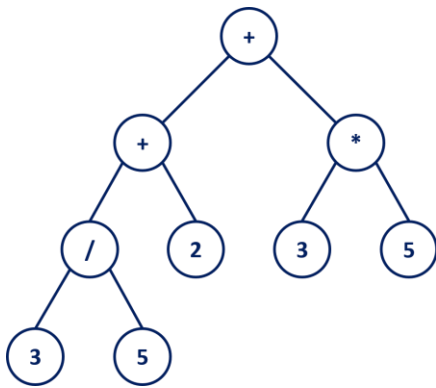
3, 7, 16, 15, 14, 4, 24, 30, 18, 48, 57, 39

3) Considerando as expressões aritméticas:

- $3 / 5 + 2 + (3 * 5)$
- $2 * 7 + 2 / 1 * 5 * 1 / 2$
- $A / B / C / D + E / F$
- $2 + 5 * 4 - 2 * 2 + 2$

Mostre as árvores binárias que as representam e as notações correspondentes (prefixa e pós-fixa) em cada caso.

a) $3 / 5 + 2 + (3 * 5)$



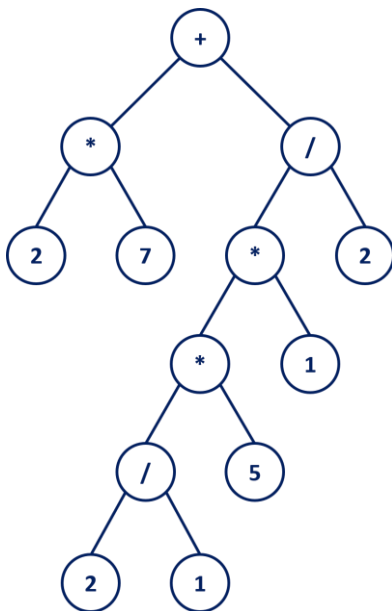
• Prefixa:

$+, +, /, 3, 5, 2, *, 3, 5$

• Pós-fixa:

$3, 5, /, 2, +, 3, 5, *, +$

b) $2 * 7 + 2 / 1 * 5 * 1 / 2$



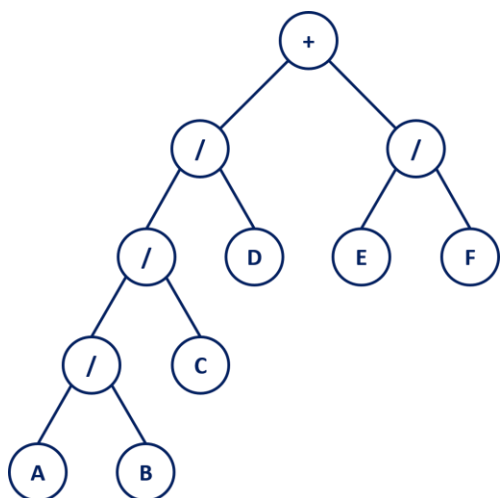
• Prefixa:

$+, *, 2, 7, /, *, *, /, 2, 1, 5, 1, 2$

• Pós-fixa:

$2, 7, *, 2, 1, /, 5, *, 1, *, 2, /, +$

c) $A / B / C / D + E / F$



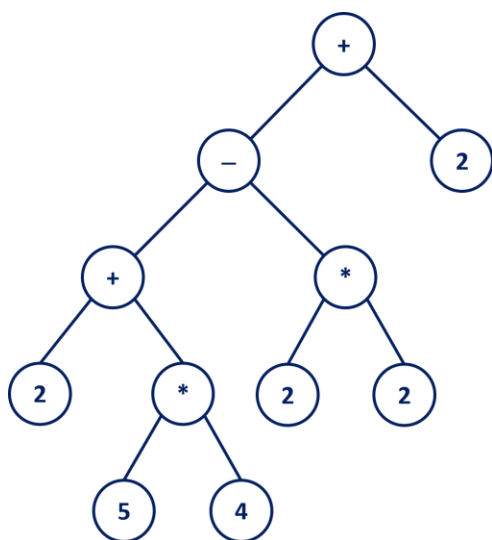
• **Prefixa:**

$+$, $/$, $/$, $/$, A , B , C , D , $/$, E , F

• **Pós-fixa:**

A , B , $/$, C , $/$, D , $/$, E , F , $/$, $+$

d) $2 + 5 * 4 - 2 * 2 + 2$



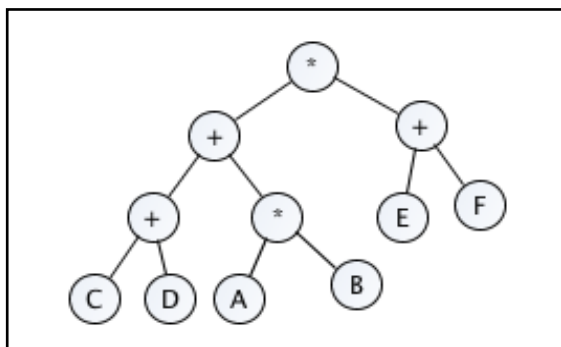
• **Prefixa:**

$+$, $-$, $+$, 2 , $*$, 5 , 4 , $*$, 2 , 2 , 2

• **Pós-fixa:**

2 , 5 , 4 , $*$, $+$, 2 , 2 , $*$, $-$, 2 , $+$

4) Considerando a árvore binária, mostrada na figura abaixo, escreva os percursos na pré-ordem, em ordem e pós-ordem, respectivamente.



• **Pré-ordem:** $*$, $+$, $+$, C , D , $*$, A , B , $+$, E , F

• **Em ordem:** C , $+$, D , $+$, A , $*$, B , $*$, E , $+$, F

• **Pós-ordem:** C , D , $+$, A , B , $*$, $+$, E , F , $+$, $*$

- 5) Considerando o modelo matemático, abaixo, construa a função C++ correspondente e a pilha de recursão para calcular o valor de $mt(2,5)$.

$$mt(a, b) = \begin{cases} a, & \text{se } b = 1 \\ mt(a, b - 1) + 1, & \text{se } b > 1 \end{cases}$$

recursiva.h

```
#ifndef pilha_recursiva
#define recursiva

double mt(int a, int b)
{
    if (b == 1)
    {
        return a;
    }
    else
    {
        return mt(a, b - 1) + 1;
    }
}

#endif // recursiva
```

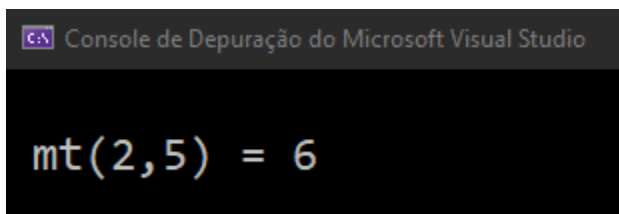
main.cpp

```
#include<iostream>
#include"recursiva.h"

using namespace std;

int main()
{
    cout << "mt(2,5) = " << mt(2,5) << endl;

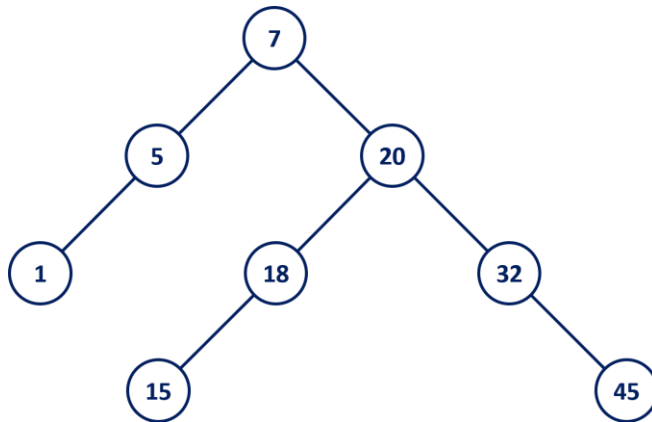
    return 0;
}
```



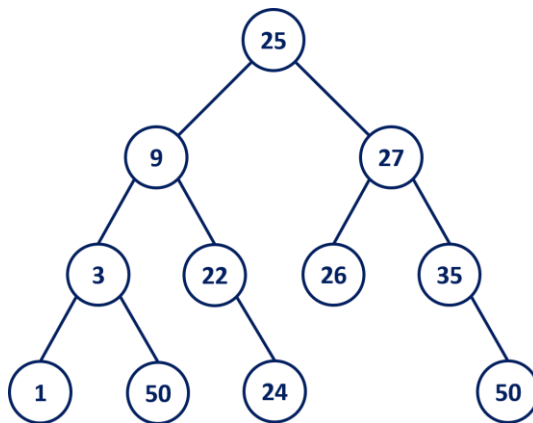
$mt(2,5) \rightarrow mt(2,4) + 1 \rightarrow (mt(2,3) + 1) + 1 \rightarrow ((mt(2,2) + 1) + 1) + 1 \rightarrow (((mt(2,1) + 1) + 1) + 1) + 1 \rightarrow$
 $mt(2,1) + 4 \rightarrow 2 + 4 \rightarrow 6$

6) Considerando os números inteiros, abaixo, construa a árvore AVL, inicialmente vazia, nos seguintes casos:

a) 1 , 5 , 7 , 8 , 15 , 18 , 20 , 23 , 32 , 45 , remove(8) , remove(23)



b) 3 , 5 , 19 , 2 , 1 , 27 , 10 , 22 , 35 , 25 , 26 , 24 , 50 , 32 , remove (19) , remove(10) , remove(32)



7) Insira em uma Hash Table de largura 3, os seguintes números inteiros: 10, 2, 15, 5, 9, 19, 6, 11, 14, 17, 13, 24, 1, 4, 20, 18, 7, 12, 21, 22 e verifique se a função de espalhamento $(x \% 3)$ é ótima.

$h[0] = 15 \rightarrow 9 \rightarrow 6 \rightarrow 24 \rightarrow 18 \rightarrow 12 \rightarrow 21$

$h[1] = 10 \rightarrow 19 \rightarrow 13 \rightarrow 1 \rightarrow 4 \rightarrow 7 \rightarrow 22$

$h[2] = 2 \rightarrow 5 \rightarrow 11 \rightarrow 14 \rightarrow 17 \rightarrow 20$

O espalhamento $(x\%3)$ é ótimo.