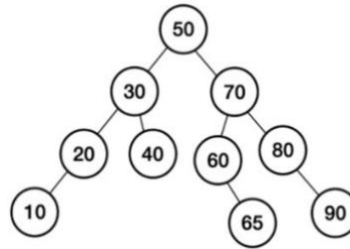


ÁRVORES EM PYTHON

1- Seja a seguinte árvore binária de busca, marque a alternativa correta:



Resposta: Todos os nós da árvore binária estão corretamente dispostos na árvore, respeitando as regras conceituais de árvores binárias de busca.

2- Seja a função de percurso in-ordem em python. Marque a opção que representa a complexidade de execução:

Resposta: A complexidade computacional do algoritmo para percurso em ordem simétrica é $O(n)$.

3- Seja o seguinte código em python cujo principal objetivo é implementar uma árvore binária. Marque a alternativa correta quanto a execução do código:

```
class NoArvore:
    def __init__(self, chave = None, esquerda = None, direita = None):
        self.chave = chave
        self.esquerda = esquerda
        self.direita = direita

if __name__ == '__main__':
    raiz = NoArvore(55)

    raiz.esquerda = NoArvore(35)
    raiz.direita = NoArvore(75)

    raiz.direita.esquerda = NoArvore(65)
    raiz.direita.direita = NoArvore(85)
    raiz.esquerda.esquerda = NoArvore(25)
    raiz.esquerda.direita = NoArvore(45)
```

Resposta: A árvore criada no código acima é uma árvore binária de busca com todas as folhas no último nível.

4- Seja a seguinte função funcaoBST em python que executa uma operação em uma árvore binária de busca.

```
def funcaoBST (raiz, chave):
    if raiz is None:
        return NoArvore(chave)
    else:
        if raiz.chave == chave:
            return raiz
        elif raiz.chave < chave:
            raiz.direita = funcaoBST(raiz.direita, chave)
        else:
            raiz.esquerda = funcaoBST(raiz.esquerda, chave)
    return raiz
```

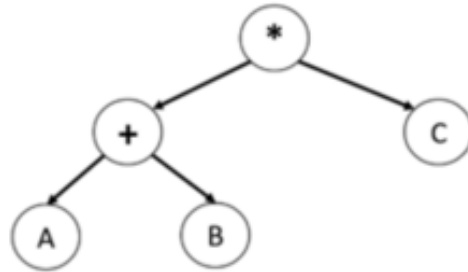
O que é executado na função acima é:

Resposta: Inserção de chave em árvore binária de busca.

5- Seja a expressão aritmética infix $A + B * C$. A sua representação posfixa é:

Resposta: $B C * A +$

6- Seja a seguinte árvore de expressões aritméticas:



A expressão aritmética que é representada pela árvore acima é:

Resposta: $(A + B) * C$