Os algoritmos são utilizados em diversos programas de computador para auxiliar no seu funcionamento correto. Dentre suas principais características temos:

R: Encerramento garantido

Um vetor ou array é uma estrutura de dados simples que armazena elementos sequencialmente em memória. O tamanho em memória necessário para armazenar um vetor de 34 elementos onde cada elemento é uma variável inteira que ocupa 2 bytes é:

R: 8 bytes.

Considerando que em uma estrutura do tipo lista circular simplesmente encadeada e com nó cabeça, a inserção ocorre sempre no início da lista, quais são os passos para realizar a inserção de um novo nó?

R: Apontar o novo nó para o seguinte ao nó cabeça, apontar o nó cabeça para o novo nó.

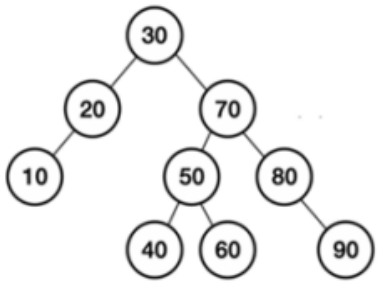
Uma Fila é uma estrutura de dados que permite o armazenamento de elementos (ou nós) sequencialmente. Sobre as Filas é possível afirmar que:

R:Permitem inserção no seu final e remoção apenas no seu início.

Uma Deque é uma estrutura de dados que permite o armazenamento de elementos (ou nós) sequencialmente. Sobre as Deques é possível afirmar que:

R: Permitem inserção ou remoção apenas no seu início ou no seu final.

Seja a seguinte árvore AVL abaixo. Com a inserção da chave 65, marque a opção que indica exatamente o que acontecerá com a árvore resultante após essa inserção:

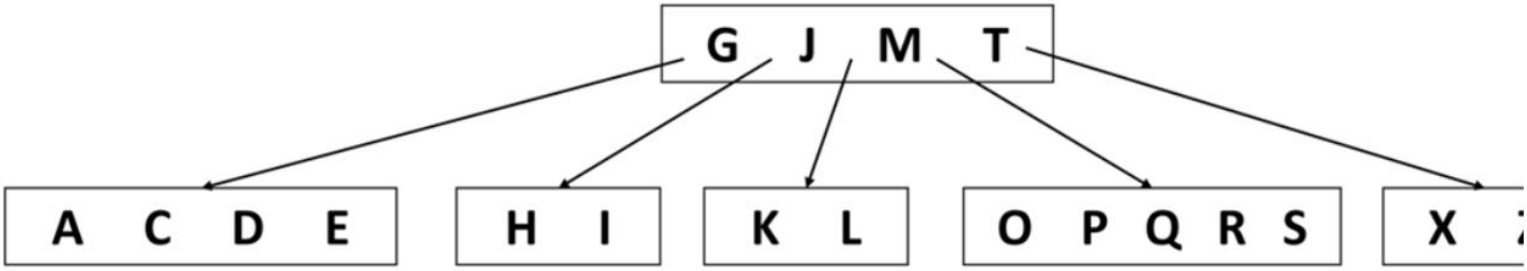


- ☐ Irá desbalancear o nó raiz da árvore AVL.
- ☐✔ Irá desbalancear o nó 30 à direita.
- ☐ O 65 será inserido à esquerda do nó 80, não causando desbalanceamento.
- ☒✘ Irá desbalancear o nó 70 à esquerda.
- ☐ Irá desbalancear o nó 20 à direita.

As árvores binárias de busca são especializações das árvores binárias que permitem uma melhor organização dos algoritmos de busca. Sobre a inserção de uma nova chave em uma árvore binária de busca é correto afirmar que:

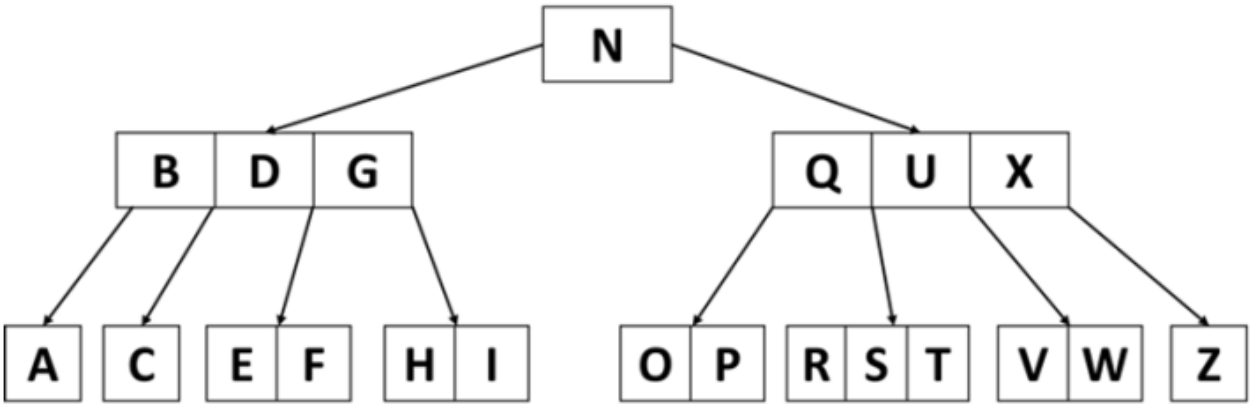
R:Todas as chaves são inseridas em folhas, a posição da folha é determinada pela busca.

Seja a seguinte árvore B de ordem n=5. O que acontecerá nesta árvore após a inserção de uma chave de valor B? Marque a opção correta:



☐✔ A chave B será inserida após a chave A.

Seja a operação de busca de chaves em uma Árvore B. Na seguinte árvore B abaixo, o resultado da sequência de chaves visitadas até encontrar a chave **S** é:



☒ N-Q-S.

Sobre as árvores binárias de busca balanceadas, analise as afirmativas abaixo:
I - Tem altura proporcional a log n.
II - As árvores completas são balanceadas.
III - Existe algoritmo capaz de transformar uma árvore binária de busca não balanceada em balanceada em O(n).
IV - Toda árvore balanceada é completa.
V - A busca ocorre em um tempo proporcional a log n nas árvores balanceadas.
R: I, II, III e V são corretas.

Seja a expressão aritmética infixa A + B * C. A sua representação posfixa é:
R: B C * A +

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)
```

- ☐ A árvore criada no código é binária de busca com altura 6, isto é, com 6 níveis distintos.
- ☐ A árvore criada no código acima não é binária de busca.
- ☐ A classe NotArvore implementa regras que garantem que os nós inseridos respeitam a ordem de inserção dos nós (maiores a direita e menores a esquerda).
- ☐ Não é possível inferir a topologia da árvore com base no código.
- ☒ A árvore criada no código acima é uma árvore binária de busca com todas as folhas no último nível.

```
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
```

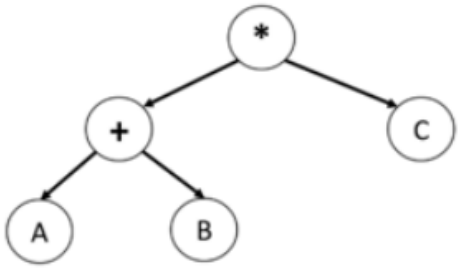
O que é executado na função acima é:

- [ ] ✔ Inserção de chave em árvore binária de busca.

Você deve implementar a operação de remoção de uma pilha (Pop), alocada contiguamente em memória, em Python. A variável da pilha é P e a próxima posição vazia da pilha é guardada pelo índice topo. Qual código dentre os seguintes realiza a implementação de forma correta?

- [ ]
```
if topo==0:
    topo=topo-1
    return P[topo]
```
- [ ]
```
if topo>0:
    topo=topo+1
    return P[topo]
```
- [x] ✔
```
if topo>0:
    topo=topo-1
    return P[topo]
```
- [ ]
```
if topo<0:
```

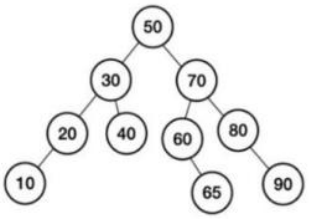
Seja a seguinte árvore de expressões aritméticas:



A expressão aritmética que é representada pela árvore acima é:

- [ ] C * (A + B)
- [ ] A + (B * C)
- [ ] A + B * C
- [x] ✔ (A + B) * C
- [ ] A + C * B

4- Seja a seguinte arvore AVL abaixo. Com a inserção da chave 65, marque a opção que indica exatamente o que acontecera com a arvore resultante após essa inserção:



Resposta: Irá desbalancear o nó 30 à direita.

Árvores de busca são estruturas de dados que permitem armazenar e recuperar informações de maneira eficiente. Marque a opção correta sobre árvores perfeitamente balanceadas
R: Toda árvore perfeitamente balanceada tem altura proporcional a log n

As árvores binárias de busca são especializações das árvores binárias que permitem uma melhor organização dos algoritmos de busca. Sobre a inserção de uma nova chave em uma árvore binária de busca é correto afirmar que:
R: Todas as chaves são inseridas em folhas, a posição da folha é determinada pela busca.

Seja a função de percurso in-ordem em Python. Marque a opção que apresenta a complexidade de execução:
R:A complexidade computacional do algoritmo para percurso em ordem simétrica é O(n).

Uma lista é uma estrutura de dados simples, que permite o armazenamento de elementos (ou nós) sequencialmente. Sobre as Listas é possível afirmar que:
R: Permitem inserção ou remoção em qualquer de suas posições.

- Uma Fila é uma estrutura de dados que permite o armazenamento de elementos (ou nós) sequencialmente. Sobre as Filas é possível afirmar que:
R: Permitem inserção no seu final e remoção apenas no seu início

Considerando que uma estrutura do tipo lista circula simplesmente encadeada e com nó cabeça, a inserção ocorre sempre ao final da lista, quais são os passos para realizar a inserção de um novo nó.
R: Percorre a lista até o último nó, apontar o ultimo nó para o novo nó, apontar o novo nó para o nó cabeça.

Suponha que você está implementando um programa que precisa armazenar dados ordenados em uma lista, que pode precisar ser percorrida em ordem crescente ou em ordem decrescente de suas chaves durante a execução do programa. A quantidade de nós durante a execução não pode ser prevista a tem o potencial de variar muito entre execuções. Qual tipo de estrutura de dados é a melhor nessa situação:
R: Lista duplamente encadeada