Linked list v stack v queue v tree v graph

**Stack:**

**Stack implementation using linked list:**

class Stack:

    def \_\_init\_\_(self):

        self.head = None

        self.num\_elements = 0

**Creating and initializing a stack using arrays:**

Class stack:

Def \_\_init\_\_(self,initial\_size = 10):

Self.arr = [0 \* for \_ in range(initial\_size)]

Self.next\_index = 0

Self.num\_elements = 0

**Queue:**

**Using arrays:**

Class Queue:

Def \_\_init\_\_(self,initial\_size = 10):

Self.arr = [0 for \_ in range(initial\_size)]

Self.next\_index = 0

Self.front\_index = -1

Self.queue\_size = 0

**Using Linked list:**

class Queue:

    def \_\_init\_\_(self):

        self.head = None

        self.tail = None

        self.num\_elements = 0

**Singly Linked list:**

Class node:

Def \_\_init\_\_(self,value):

Self.value = value

Self.next = None

class SLL:

     def \_\_init\_\_(self):

         self.head = None

**Doubly Linked List:**

Class doubleNode:

            Def \_\_init\_\_(self, value):

                        Self.value = value

                        Self.next = None

                        Self.prev = None

Class doublyLinkedList:

            Def \_\_init\_\_(self):

                        Self.head = None

                        Self.tail = None

**Tree:**

class Tree(object):

    def \_\_init\_\_(self,value):

        self.root = Node(value)

    def get\_root(self):

        return self.root

class Node:

    def \_\_init\_\_(self,value=None):

        self.value = value

        self.left = None

        self.right = None