Homework 3

David Trinh

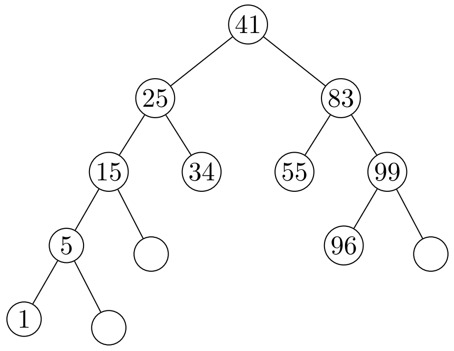
October 1, 2024

- **Question 1**

```
def getTopology(A):
    n = len(A)
    isRing = True
    isStar = True
    isFullyConnectedMesh = True
    isCentralNode = False

    for i in range(n):
        totalAdjacent = 0
        for j in range(n):
            if A[i][j]:
                totalAdjacent += 1

        if isFullyConnectedMesh and totalAdjacent != n - 1:
            isFullyConnectedMesh = False
        if isStar:
            if totalAdjacent == n - 1:
                isCentralNode = True
            elif totalAdjacent != 1:
                isStar = False
        if isRing and totalAdjacent != 2:
            isRing = False

    if isRing:
        return "Ring"
    elif isFullyConnectedMesh:
        return "Fully Connected Mesh"
    elif isStar and isCentralNode:
        return "Star"
    else:
        return "None of the above"
```

- **Question 2**

1. Binary Search Tree



2. Pre-order traversal

   41, 25, 15, 5, 1, 34, 83, 55, 99, 96

3. In-order traversal

   1, 5, 15, 25, 34, 41, 55, 83, 96, 99

4. Post-order traversal

   1, 5, 15, 34, 25, 55, 96, 99, 83, 41

- **Question 3**

```
def insert(self, newValue):
    if self.isEmpty():
        self.emptyTree = False
        self.setNodeValue(newValue)
        return

    if newValue == self.getNodeValue():
        return

    if newValue < self.getNodeValue():
        if self.hasLeftChild():
            self.getLeftChild().insert(newValue)
        else:
            self.setLeftChild(BST(newValue))
    else:
        if self.hasRightChild():
            self.getRightChild().insert(newValue)
        else:
            self.setRightChild(BST(newValue))
```

n	1	10	100	1000	10000	100000	1000000	10000000	100000000
add to front of list	2	1	1	1	3	22	too big	too big	too big
add to middle of list	1	1	1	1	2	12	too big	too big	too big
add to end of list	1	1	1	1	1	1	too big	too big	too big
del from front of list	1	0	0	0	1	7	too big	too big	too big
del from middle of list	0	0	0	0	0	4	too big	too big	too big
del from end of list	0	0	0	0	0	0	too big	too big	too big

Unit is in microsecond (0.000001s)

I expected that adding and deleting items to the end of the list are faster than the middle or the front, considering the fact that python implements lists using array. According to the table, this hypothesis holds up as large lists still perform adding and deleting at 1 or less microseconds, which process averages O(1).