AI Assignment

Submitted BY:Ahmad Hasan Syed
Roll NO: p18-0126

Google drive link of video:
https://drive.google.com/file/d/1yE3d64z03-7CrpYr0KjSQYUsY6fhtWjc/view?usp=sharing

Google meet was not showing the screen sometimes so I have attached the code below!

```
s=[[3,4,1,3,1],
   [3,3,3,'G',2],
   [3,1,2,2,3],
   [4,2,3,3,3],
   [4,1,4,3,2]

]

move_check=0
def move():
    if move_check==0:
        i=0
        j=0

        move_check+=1
        for i in range(1):
                i=i+1
                return (s[i][j])

    if move_check==1:
        i=1
        j=-1
        move_check+=1
        while j<5:
            j=j+1
            return (s[i][j])
    if move_check ==2:
        i=2
        j=-1
        move_check+=1
        while j<5:
            j=j+1
```

```
            return (s[i][j])

    if move_check ==3:
        i=3
        j=-1
        move_check+=1
        while j<5:
            j=j+1
            return (s[i][j])

    if move_check==4:
        i=4
        j=-1
        move_check+=1
        while j<5:
            j=j+1
            return (s[i][j])
class Node:


    def __init__(self, val):
        self.l = None
        self.r = None
        self.v = val



class Tree:
    def __init__(self):
        self.root = None

    def getRoot(self):
        return self.root
```

```
def add(self, val):
    if(self.root == None):
        self.root = Node(move())
    else:
        self._add(move(), self.root)


def _add(self, val, node):

        if(node.l != None):
            self._add(move(), node.l)
        if(node.l==None):
            node.l = Node(move())

        if(node.r != None):
            self._add(move(), node.r)
        if(node.r==None):
            node.r = Node(move())


def printTree(self):
    if(self.root != None):
        self._printTree(self.root)


def _printTree(self, node):
    if(node != None):
        self._printTree(node.l)
        print(str(node.v) + ' ')
        self._printTree(node.r)
```

```
    def bfs(self, graph, start):
        visited, queue = set(), [start]
        while queue:
            vertex = queue.pop()
            if vertex!='G':  #goal state found!!
                if vertex not in visited:
                    visited.add(vertex)
                    # new nodes are added to end of queue
                    queue.extend(graph[vertex] - visited)
            return visited #if goad state visited then return the
```
list! and the path cost will be counted on the basis of list len() function!